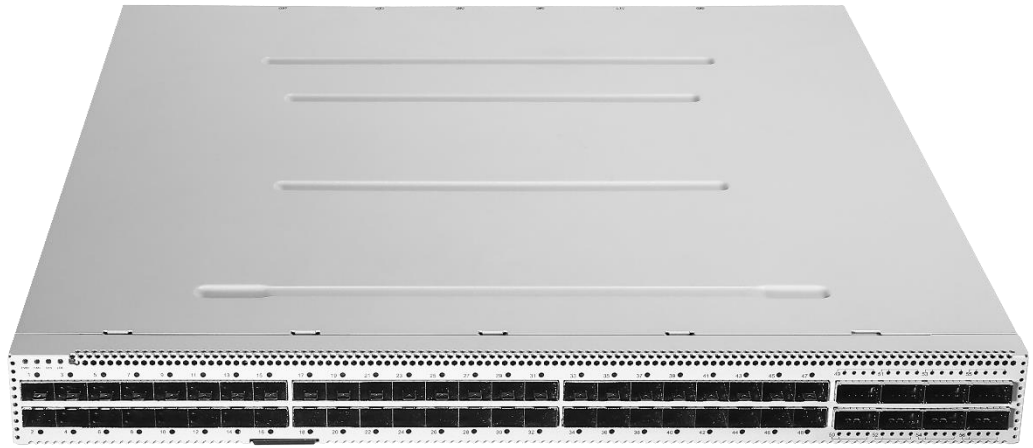




CLI Reference Guide

Micro Data Center Switch DQS-5000-56ZS SONiC OS



Contents

1	Configuration	3
1.1	SONiC Configuration Database Manual	3
1.2	Introduction	3
1.3	Configuration	3
1.4	Config Load and Save	3
1.5	Redis and Json Schema	3
1.6	For Developers	54
1.6.1	Generating Application Config by Jinja2 Template	54
1.6.2	Incremental Configuration by Subscribing to ConfigDB	54
2	Command reference	54
2.1	SONiC Command Line Interface Guide	54
2.1.1	Introduction	54
2.1.2	Basic Tasks	55
2.1.3	Getting Help	56
2.1.4	Basic Show Commands	59
2.1.5	AAA & TACACS+	79
2.1.6	ACL	84
2.1.7	ASIC SDK health event	89
2.1.8	ARP & NDP	92
2.1.9	BFD	93
2.1.10	BGP	94
2.1.11	Console	104
2.1.12	CMIS firmware upgrade	108
2.1.13	CMIS debug	111
2.1.14	DHCP Relay	111
2.1.15	Drop Counters	115
2.1.16	Dynamic Buffer Management	118
2.1.17	Feature	125
2.1.18	Flow Counters	128
2.1.19	Gearbox	131
2.1.20	Update Device Hostname Configuration Commands	131
2.1.21	Hash	132
2.1.22	Interfaces	136
2.1.23	Interface Naming Mode	159
2.1.24	Interface Vrf binding	161
2.1.25	IP / IPv6	161
2.1.26	IPv6 Link Local	166
2.1.27	Kubernetes	168
2.1.28	LDAP	169
2.1.29	Linux Kernel Dump	170
2.1.30	LLDP	174



2.1.31 Loading, Reloading And Saving Configuration	176
2.1.32 Loopback Interfaces	179
2.1.33 VRF Configuration	180
2.1.34 Management VRF	181
2.1.35 Muxcable	185
2.1.36 Mirroring	192
2.1.37 NAT	195
2.1.38 NTP	201
2.2 PFC Watchdog Commands	202
2.2.1 Platform Component Firmware	203
2.2.2 Platform Specific Commands	209
2.2.3 PortChannels	210
2.2.4 NVGRE	211
2.2.5 PBH	213
2.2.6 PINS	218
2.2.7 QoS	219
2.2.8 Radius	229
2.2.9 sFlow	230
2.2.10 SNMP	234
2.2.11 Startup & Running Configuration	239
2.2.12 Static routing	242
2.2.13 Subinterfaces	243
2.2.14 Switchport Modes	244
2.2.15 Syslog	245
2.2.16 System State	250
2.2.17 VLAN & FDB	258
2.2.18 Troubleshooting Commands	284
2.2.19 Routing Stack	290
2.2.20 Quagga BGP Show Commands	290
2.2.21 ZTP Configuration and Show Commands	294
2.3 SFP Utilities Commands	297
2.3.1 SFP Utilities show commands	297
2.4 SFP Utilities read command	300
2.5 SFP Utilities write command	300
2.6 Static DNS Commands	301
2.6.1 Static DNS config command	301
2.6.2 Static DNS show command	301
2.7 Wake-on-LAN Commands	302
2.7.1 Send Wake-on-LAN Magic Packet command	302
2.8 Banner Commands	302
2.8.1 Banner config commands	302
2.8.2 Banner show command	303
2.9 Memory Statistics Commands	304
2.9.1 Overview	304
2.9.2 Memory Statistics Config Commands	304
2.9.3 Memory Statistics Show Commands	305



1 Configuration

1.1 SONiC Configuration Database Manual

1.2 Introduction

This document lists the configuration commands schema applied in the SONiC eco system. All these commands find relevance in collecting system information, analysis and even for trouble shooting. All the commands are categorized under relevant topics with corresponding examples.

1.3 Configuration

SONiC is managing configuration in a single source of truth - a redisDB instance that we refer as ConfigDB. Applications subscribe to ConfigDB and generate their running configuration correspondingly.

1.4 Config Load and Save

In current version of SONiC, ConfigDB is implemented as database 4 of local redis. When system boots, configurations will be loaded from `/etc/sonic/config_db.json` file into redis. Please note that ConfigDB content won't be written back into `/etc/sonic/config_db.json` file automatically. In order to do that, a config save command need to be manually executed from CLI. Similarly, config load will trigger a force load of json file into DB. Generally, content in `/etc/sonic/config_db.json` can be considered as starting config, and content in redisDB running config.

We keep a way to load configuration from minigraph and write into ConfigDB for backward compatibility. To do that, run `config load_minigraph`.

1.4.0.1 Incremental Configuration

The design of ConfigDB supports incremental configuration - application could subscribe to changes in ConfigDB and response correspondingly. The only application that supports incremental configuration is BGP (docker-fpm-quagga). For other applications, a manual restart is required after configuration changes in ConfigDB.

1.5 Redis and Json Schema

ConfigDB uses a table-object schema that is similar with AppDB, and `config_db.json` is a straight-forward serialization of DB. As an example, the following fragments could be BGP-related configuration in redis and json, correspondingly:

Redis format

```
127.0.0.1:6379[4]> keys BGP_NEIGHBOR|*
```

- 1) "BGP_NEIGHBOR|10.0.0.31"
- 2) "BGP_NEIGHBOR|10.0.0.39"
- 3) "BGP_NEIGHBOR|10.0.0.11"
- 4) "BGP_NEIGHBOR|10.0.0.7"

...



```
127.0.0.1:6379[4]> hgetall BGP_NEIGHBOR|10.0.0.3
1) "admin_status"
2) "up"
3) "peer_addr"
4) "10.0.0.2"
5) "asn"
6) "65200"
7) "name"
8) "DQS5K07T2"
```

Json format

```
"BGP_NEIGHBOR":
{
  "10.0.0.57":
  {
    "rrclient": "0", "name":
    "DQS5K01T1",
    "local_addr": "10.0.0.56",
    "nhopself": "0", "holdtime": "10",
    "asn": "64600",
    "keepalive": "3"
  },
  "10.0.0.59":
  {
    "rrclient": "0", "name":
    "DQS5K02T1",
    "local_addr": "10.0.0.58",
    "nhopself": "0", "holdtime": "10",
    "asn": "64600",
    "keepalive": "3"
  },
}
```

1.5.0.1 ACL and Mirroring

ACL and mirroring related configuration are defined in **MIRROR_SESSION**, **ACL_TABLE** and **ACL_RULE** tables. Those tables are in progress of migrating from APPDB. Please refer to their schema in APPDB [here](#) and migration plan [here](#).

```
{ "MIRROR_SESSION":
  { "everflow0":
    { "src_ip": "192.168.100.66",
      "dst_ip": "192.168.101.88" }
  },
  "ACL_TABLE":
  { "DATAACL":
    { "policy_desc": "data_acl",
      "type": "l3",
      "ports": [ "Ethernet2", "Ethernet4", "Ethernet4", "Ethernet8" ] }
  }
}
```



Below ACL table added as per the mail

```
{
  "ACL_TABLE":
    { "aaa": { "type": "L3", "ports": "Ethernet0" } },
  "ACL_RULE":
    { "aaa|rule_0":
      { "PRIORITY": "55", "PACKET_ACTION": "DROP", "L4_SRC_PORT": "0"},
      "aaa|rule_1":
        { "PRIORITY": "55", "PACKET_ACTION": "DROP", "L4_SRC_PORT": "1" } } }
```

Below ACL table added by comparig minigraph.xml & config_db.json

```
{
  "ACL_TABLE":
    { "EVERFLOW":
      { "type": "MIRROR",
        "policy_desc": "EVERFLOW",
        "ports": [ "PortChannel0001", "PortChannel0002", "PortChannel0003", "PortChannel0004" ] },
      "EVERFLOWV6":
        { "type": "MIRRORV6",
          "policy_desc": "EVERFLOWV6",
          "ports": [ "PortChannel0001", "PortChannel0002", "PortChannel0003", "PortChannel0004" ] },
      "SNMP_ACL":
        { "services": [ "SNMP" ],
          "type": "CTRLPLANE",
          "policy_desc": "SNMP_ACL"},
      "SSH_ONLY":
        { "services": [ "SSH" ],
          "type": "CTRLPLANE",
          "policy_desc": "SSH_ONLY"}},
  "ACL_RULE":
    { "SNMP_ACL|DEFAULT_RULE": { "PRIORITY": "1", "PACKET_ACTION": "DROP", "ETHER_TYPE": "2048"},
      "SNMP_ACL|RULE_1": { "PRIORITY": "8888", "PACKET_ACTION": "ACCEPT", "SRC_IP": "10.10.10.10/32",
        "IP_PROTOCOL": "17"},
      "SNMP_ACL|RULE_2": { "PRIORITY": "8887", "PACKET_ACTION": "ACCEPT", "SRC_IP": "20.20.20.20/32",
        "IP_PROTOCOL": "17"},
      "SSH_ONLY|DEFAULT_RULE": { "PRIORITY": "1", "PACKET_ACTION": "DROP", "ETHER_TYPE": "2048"},
      "SSH_ONLY|RULE_1": { "PRIORITY": "8888", "PACKET_ACTION": "ACCEPT", "SRC_IP": "40.40.40.40/8",
        "IP_PROTOCOL": "6" }
    }
}
```



ACL table type configuration example

```
{
  "ACL_TABLE_TYPE":
  {
    "CUSTOM_L3":
    { "MATCHES":
      [
        "IN_PORTS",
        "OUT_PORTS",
        "SRC_IP",
        "TUNNEL_TERM"],
      "ACTIONS":
      [
        "PACKET_ACTION",
        "MIRROR_INGRESS_ACTION"],
      "BIND_POINTS":
      [
        "PORT",
        "PORTCHANNEL"
      ]
    },
    "ACL_TABLE":
    {
      "DATAACL":
      {
        "STAGE": "INGRESS",
        "TYPE": "CUSTOM_L3",
        "PORTS":
        [
          "Ethernet0",
          "PortChannel1"
        ]
      }
    },
    "ACL_RULE":
    {
      "DATAACL|RULE0":
      {
        "PRIORITY": "9999",
        "PACKET_ACTION": "DROP",
        "SRC_IP": "10.10.10.10/32",
        "TUNNEL_TERM": "true"
      }
    }
  }
}
```

1.5.0.2 BGP BBR

The **BGP_BBR** table contains device-level BBR state.

```
{
  "BGP_BBR":
  {
    "all":
    { "status": "enabled"/"disabled" }
  }
}
```



1.5.0.3 ASIC SDK health event

ASIC/SDK health event related configuration is defined in **SUPPRESS_ASIC_SDK_HEALTH_EVENT** table.

```
"SUPPRESS_ASIC_SDK_HEALTH_EVENT":
{
  "notice":
  {
    "categories": ["asic_hw"],
    "max_events": "1000" },
  "warning":
  {
    "categories": ["software", "cpu_hw"]
  }
}
```

1.5.0.4 BGP Device Global

The **BGP_DEVICE_GLOBAL** table contains device-level BGP global state.

It has a STAT object containing device state like **tsa_enabled**, **wcmp_enabled** and **idf_isolation_state**.

When **tsa_enabled** is set to true, the device is isolated using **traffic-shift-away (TSA)** route-maps in BGP.

```
{
  "BGP_DEVICE_GLOBAL":
  {
    "STATE":{"tsa_enabled": "true"}
  }
}
```

When **wcmp_enabled** is set to true, the device is configured to use **BGP Link Bandwidth Extended Community**.

Weighted ECMP load balances traffic between the equal cost paths in proportion to the capacity of the local links.

```
{
  "BGP_DEVICE_GLOBAL":
  {
    "STATE": {"wcmp_enabled": "true" }
  }
}
```

The IDF isolation state **idf_isolation_state** could be one of **isolated_no_export**, **isolated_withdraw_all** or **unisolated**.

```
{
  "BGP_DEVICE_GLOBAL":
  {
    "STATE":
    { "idf_isolation_state": "isolated_no_export" }
  }
}
```



1.5.0.5 BGP Sessions

BGP session configuration is defined in **BGP_NEIGHBOR** table. BGP neighbor address is used as key of bgp neighbor objects. Object attributes include remote AS number, neighbor router name, and local peering address. Dynamic neighbor is also supported by defining peer group name and IP ranges in **BGP_PEER_RANGE** table.

```
{
  "BGP_NEIGHBOR":
  {
    "10.10.10.61":
      { "local_addr": "10.10.10.60", "asn": 64010, "name": "DQS5K15T0" },
    "10.10.10.31":
      { "local_addr": "10.0.0.30", "asn": 64008, "name": "DQS5K09T0" },
    "10.10.10.21":
      { "rrclient": "0", "name": "DQS5K04T1", "local_addr": "10.0.0.20", "nhopself": "0", "holdtime": "10",
        "asn": "64800", "keepalive": "3" }
  }
  "BGP_PEER_RANGE":
  { "BGPSLBPassive":
    { "name": "BGPSLBPassive", "ip_range": ["10.100.0.0/27" ] },
    "BGPVac":
    { "name": "BGPVac", "ip_range": ["10.200.0.0/16" ] },
    "BGPSentinel":
    { "name": "BGPSentinel", "ip_range": ["10.10.0.0/24" ], "src_address": "10.10.0.32" }
  }
}
```



1.5.0.6 BUFFER_PG

When the system is running in **traditional buffer model**, profiles needs to explicitly configured:

```
{ "BUFFER_PG":
  { "Ethernet0|3-4":
    {
      "profile": "pg_lossless_40000_5m_profile" },
      "Ethernet1|3-4": {"profile": "pg_lossless_40000_5m_profile" },
      "Ethernet2|3-4": {"profile": "pg_lossless_40000_5m_profile" }
    }
  }
}
```

When the system is running in **dynamic buffer model**, profiles can be:

- either calculated dynamically according to ports' configuration and just configured as "NULL";
- or configured explicitly.

```
{
  "BUFFER_PG":
  {
    "Ethernet0|3-4": {"profile": "NULL" },
    "Ethernet1|3-4": {"profile": "NULL"},
    "Ethernet2|3-4": {"profile": "static_profile" }
  }
}
```

1.5.0.7 Buffer pool

When the system is running in traditional buffer model, the size of all of the buffer pools and xoff of ingress_lossless_pool need to be configured explicitly.

```
{
  "BUFFER_POOL":
  {
    "egress_lossless_pool":
    {
      "type": "egress",
      "mode": "static",
      "size": "14281831"
    },
    "egress_lossy_pool":
    {
      "type": "egress",
      "mode": "dynamic",
      "size": "7163815"
    },
    "ingress_lossless_pool":
    {
      "xoff": "2102715",
      "type": "ingress",
      "mode": "dynamic",
      "size": "11235054"
    }
  }
}
```



When the system is running in dynamic buffer model, the size of some of the buffer pools can be omitted and will be dynamically calculated. In this case, A percentage can be configured on a pool, representing how many the available buffer can be allocated to the pool.

```
{
  "BUFFER_POOL":
  {
    "egress_lossless_pool":
    {
      "type": "egress",
      "mode": "static",
      "size": "14332621"
    },
    "egress_lossy_pool":
    {
      "type": "egress",
      "mode": "dynamic"
    },
    "ingress_lossless_pool":
    {
      "type": "ingress",
      "mode": "dynamic",
      "percentage": "75"
    }
  }
}
```

1.5.0.8 Buffer profile

```
{
  "BUFFER_PROFILE":
  {
    "egress_lossless_profile":
    { "static_th": "2885660",
      "pool": "egress_lossless_pool",
      "size": "1518" },
    "egress_lossy_profile":
    { "dynamic_th": "3",
      "pool": "egress_lossy_pool",
      "size": "1518" },
    "ingress_lossy_profile":
    { "dynamic_th": "3",
      "pool": "ingress_lossless_pool",
      "size": "0" },
    "pg_lossless_40000_10m_profile":
    { "xon_offset": "2288",
      "dynamic_th": "-3",
      "xon": "2288",
      "xoff": "66560",
      "pool": "ingress_lossless_pool",
      "size": "1248" },
    "pg_lossless_40000_30m_profile":
    { "xon_offset": "2288",
      "dynamic_th": "-3",
      "xon": "2288",
      "xoff": "71552",
      "pool": "ingress_lossless_pool",
      "size": "1248"}
  }
}
```



When the system is running in dynamic buffer model and the headroom_type is dynamic, only **dynamic_th** needs to be configured and rest of fields can be omitted. This kind of profiles will be handled by buffer manager and won't be applied to SAI.

```
{
  {
    "non_default_dynamic_th_profile":
    {
      "dynamic_th": 1,
      "headroom_type": "dynamic"
    }
  }
}
```

1.5.0.9 Buffer queue

```
{
  "BUFFER_QUEUE":
  {
    "Ethernet50,Ethernet52,Ethernet54,Ethernet56|0-2":
    { "profile": "egress_lossy_profile" },
    "Ethernet50,Ethernet52,Ethernet54,Ethernet56|3-4":
    { "profile": "egress_lossless_profile" },
    "Ethernet50,Ethernet52,Ethernet54,Ethernet56|5-6":
    { "profile": "egress_lossy_profile" }
  }
}
```

1.5.0.10 Buffer port ingress profile list

```
{
  "BUFFER_PORT_INGRESS_PROFILE_LIST":
  {
    "Ethernet50":
    { "profile_list": "ingress_lossy_profile,ingress_lossless_profile" },
    "Ethernet52":
    { "profile_list": "ingress_lossy_profile,ingress_lossless_profile" },
    "Ethernet56":
    { "profile_list": "ingress_lossy_profile,ingress_lossless_profile" }
  }
}
```

1.5.0.11 Buffer port egress profile list

```
{
  "BUFFER_PORT_EGRESS_PROFILE_LIST":
  {
    "Ethernet50":
    { "profile_list": "egress_lossy_profile,egress_lossless_profile" },
    "Ethernet52":
    { "profile_list": "egress_lossy_profile,egress_lossless_profile" },
    "Ethernet56":
    { "profile_list": "egress_lossy_profile,egress_lossless_profile" }
  }
}
```




1.5.0.11 Buffer port egress profile list

```
{
  "BUFFER_PORT_EGRESS_PROFILE_LIST":
    {
      "Ethernet50": { "profile_list": "egress_lossy_profile,egress_lossless_profile" },
      "Ethernet52": { "profile_list": "egress_lossy_profile,egress_lossless_profile" },
      "Ethernet56": { "profile_list": "egress_lossy_profile,egress_lossless_profile" }
    }
}
```

1.5.0.12 Cable length

```
{
  "CABLE_LENGTH":
    { "AZURE": {
      "Ethernet8": "5m",
      "Ethernet9": "5m",
      "Ethernet2": "5m",
      "Ethernet58": "5m",
      "Ethernet59": "5m",
      "Ethernet50": "40m",
      "Ethernet51": "5m",
      "Ethernet52": "40m",
      "Ethernet53": "5m",
      "Ethernet54": "40m",
      "Ethernet55": "5m",
      "Ethernet56": "40m"
    }
  }
}
```



1.5.0.14 COPP_TABLE

```
{
  "COPP_TABLE":
    {"default":
      {
        "cbs": "600",
        "cir": "600",
        "meter_type": "packets",
        "mode": "sr_tcm",
        "queue": "0",
        "red_action": "drop"
      },
      "trap.group.arp":
        {
          "cbs": "600",
          "cir": "600",
          "meter_type": "packets",
          "mode": "sr_tcm",
          "queue": "4",
          "red_action": "drop",
          "trap_action": "trap",
          "trap_ids": "arp_req,arp_resp,neigh_discovery",
          "trap_priority": "4"
        },
      "trap.group.lldp.dhcp.udld":
        {
          "queue": "4",
          "trap_action": "trap",
          "trap_ids": "lldp,dhcp,udld",
          "trap_priority": "4"
        },
      "trap.group.bgp.lacp":
        {
          "queue": "4",
          "trap_action": "trap",
          "trap_ids": "bgp,bgpv6,lacp",
          "trap_priority": "4"
        },
      "trap.group.ip2me":
        {
          "cbs": "600",
          "cir": "600",
          "meter_type": "packets",
          "mode": "sr_tcm",
          "queue": "1",
          "red_action": "drop",
          "trap_action": "trap",
          "trap_ids": "ip2me",
          "trap_priority": "1"
        }
    }
}
```



1.5.0.15 Console

```
{
  "CONSOLE_PORT":
    {
      "1":
        {
          "baud_rate": "115200",
          "flow_control": "0",
          "remote_device": "Admin_host-1"
        },
      "2":
        {
          "baud_rate": "9600",
          "flow_control": "1"
        }
    },
  "CONSOLE_SWITCH":
    {
      "console_mgmt":
        { "enabled": "yes" }
    }
}
```

1.5.0.16 Critical Resource Monitoring (CRM)

Example,

```
{
  "CRM":
    {
      "Config":
        {
          "acl_table_threshold_type": "percentage",
          "nexthop_group_threshold_type": "percentage",
          "fdb_entry_high_threshold": "85",
          "acl_entry_threshold_type": "percentage",
          "ipv6_neighbor_low_threshold": "70",
          "nexthop_group_member_low_threshold": "70",
          "acl_group_high_threshold": "85",
          "ipv4_route_high_threshold": "85",
          "acl_counter_high_threshold": "85",
          "ipv4_route_low_threshold": "70",
          "ipv4_route_threshold_type": "percentage",
          "ipv4_neighbor_low_threshold": "70",
          "acl_group_threshold_type": "percentage",
          "ipv4_nexthop_high_threshold": "85",
          "ipv6_route_threshold_type": "percentage",
          "snat_entry_threshold_type": "percentage",
          "snat_entry_high_threshold": "85",
          "snat_entry_low_threshold": "70",
          "dnat_entry_threshold_type": "percentage",
          "dnat_entry_high_threshold": "85",
          "dnat_entry_low_threshold": "70",
          "ipmc_entry_threshold_type": "percentage",
          "ipmc_entry_high_threshold": "85",
          "ipmc_entry_low_threshold": "70"
        }
    }
}
```

Tips, Source Network Address Translation means snat; IP Multicast entry means ipmc



1.5.0.17 CRM DASH

Example,

```
{
  "CRM":
  {
    "Config":
    {
      "dash_vnet_threshold_type": "percentage",
      "dash_vnet_low_threshold": "70",
      "dash_vnet_high_threshold": "85",
      "dash_eni_threshold_type": "percentage",
      "dash_eni_low_threshold": "70",
      "dash_eni_high_threshold": "85",
      "dash_eni_ether_address_map_threshold_type": "percentage",
      "dash_eni_ether_address_map_low_threshold": "70",
      "dash_eni_ether_address_map_high_threshold": "85",
      "dash_ipv4_inbound_routing_threshold_type": "percentage",
      "dash_ipv4_inbound_routing_low_threshold": "70",
      "dash_ipv4_inbound_routing_high_threshold": "85",
      "dash_ipv6_inbound_routing_threshold_type": "percentage",
      "dash_ipv6_inbound_routing_low_threshold": "70",
      "dash_ipv6_inbound_routing_high_threshold": "85",
      "dash_ipv4_outbound_routing_threshold_type": "percentage",
      "dash_ipv4_outbound_routing_low_threshold": "70",
      "dash_ipv4_outbound_routing_high_threshold": "85",
      "dash_ipv6_outbound_routing_threshold_type": "percentage",
      "dash_ipv6_outbound_routing_low_threshold": "70",
      "dash_ipv6_outbound_routing_high_threshold": "85",
      "dash_ipv4_pa_validation_threshold_type": "percentage",
      "dash_ipv4_pa_validation_low_threshold": "70",
      "dash_ipv4_pa_validation_high_threshold": "85",
      "dash_ipv6_pa_validation_threshold_type": "percentage",
      "dash_ipv6_pa_validation_low_threshold": "70",
      "dash_ipv6_pa_validation_high_threshold": "85",
      "dash_ipv4_outbound_ca_to_pa_threshold_type": "percentage",
      "dash_ipv4_outbound_ca_to_pa_low_threshold": "70",
      "dash_ipv4_outbound_ca_to_pa_high_threshold": "85",
      "dash_ipv6_outbound_ca_to_pa_threshold_type": "percentage",
      "dash_ipv6_outbound_ca_to_pa_low_threshold": "70",
      "dash_ipv6_outbound_ca_to_pa_high_threshold": "85",
      "dash_ipv4_acl_group_threshold_type": "percentage",
      "dash_ipv4_acl_group_low_threshold": "70",
      "dash_ipv4_acl_group_high_threshold": "85",
      "dash_ipv6_acl_group_threshold_type": "percentage",
      "dash_ipv6_acl_group_low_threshold": "70",
      "dash_ipv6_acl_group_high_threshold": "85",
      "dash_ipv4_acl_rule_threshold_type": "percentage",
      "dash_ipv4_acl_rule_low_threshold": "70",
      "dash_ipv4_acl_rule_high_threshold": "85",
      "dash_ipv6_acl_rule_threshold_type": "percentage",
      "dash_ipv6_acl_rule_low_threshold": "70",
      "dash_ipv6_acl_rule_high_threshold": "85"
    }
  }
}
```



1.5.0.18 DEFAULT_LOSSLESS_BUFFER_PARAMETER

This table stores the default lossless buffer parameters for dynamic buffer calculation.

Example,

```
{
  "DEFAULT_LOSSLESS_BUFFER_PARAMETER":
    { "AZURE":
      {
        "default_dynamic_th": "0",
        "over_subscribe_ratio": "2"
      }
    }
}
```

1.5.0.19 Device Metadata

The **DEVICE_METADATA** table contains only one object named **localhost**. In this table the device metadata such as hostname, hwsku, deployment environment id and deployment type are specified. BGP local AS number is also specified in this table as current only single BGP instance is supported in SONiC.

Example,

```
{
  "DEVICE_METADATA":
    {
      "localhost":
        {
          "asic_id": "06:00.0",
          "asic_name": "marvell",
          "hwsku": "D-Link_DQS-5000-56ZS-A1",
          "default_bgp_status": "up",
          "docker_routing_config_mode": "unified",
          "hostname": "sonic-dqs5000-01",
          "platform": "x86_64-dlink_dqs_5000_56zs-r0",
          "mac": "BC:22:28:12:34:56",
          "default_pfcwd_status": "disable",
          "bgp_asn": "65100",
          "deployment_id": "1",
          "type": "ToRRouter",
          "bgp_adv_lo_prefix_as_128": "true",
          "buffer_model": "traditional",
          "yang_config_validation": "disable",
          "rack_mgmt_map": "dummy_value",
          "timezone": "Europe/Berlin",
          "bgp_router_id": "8.8.8.8"
        }
    }
}
```



1.5.0.20 Device neighbor metada

Example,

```
{
  "DEVICE_NEIGHBOR_METADATA":
  {
    "DQS5K01T1":
    {
      "cluster": " D-Link_DQS-5000-56ZS-A1",
      "lo_addr": "None",
      "mgmt_addr": "10.10.15.5",
      "hwsku": " D-Link_DQS-5000-56ZS-A1-VM",
      "type": "LeafRouter"
    },
    "DQS5K02T1":
    {
      "cluster": " D-Link_DQS-5000-56ZS-A1",
      "lo_addr": "None",
      "mgmt_addr": "10.10.15.6",
      "hwsku": " D-Link_DQS-5000-56ZS-A1-VM",
      "type": "LeafRouter"
    }
  }
}
```

1.5.0.21 DHCP_RELAY

Example,

```
{
  "DHCP_RELAY":
  {
    "dhcpv6_servers": [
      "fc02:2100::6",
      "fc02:2100::7",
      "fc02:2100::8",
      "fc02:2100::9"
    ],
    "rfc6939_support": "true",
    "interface_id": "true"
  }
}
```

1.5.0.22 BMP

BMP related configuration are defined in **bgp_neighbor_table**, **bgp_rib_in_table**, **bgp_rib_out_table** tables.

```
{
  "BMP":
  {
    "table":
    {
      {
        "bgp_neighbor_table": "true",
        "bgp_rib_in_table": "false",
        "bgp_rib_out_table": "false"
      }
    }
  }
}
```



1.5.0.23 DHCP_SERVER_IPV4

Ipv4 DHCP Server related configuration are defined in **DHCP_SERVER_IPV4**, **DHCP_SERVER_IPV4_RANGE**, **DHCP_SERVER_IPV4_CUSTOMIZED_OPTIONS**, **DHCP_SERVER_IPV4_PORT** tables.

```
{
  "DHCP_SERVER_IPV4":
  {
    "Vlan101":
    {
      "gateway": "101.1.1.1",
      "lease_time": 3600,
      "mode": "PORT",
      "netmask": "255.255.255.0",
      "customized_options": ["option60" ],
      "state": "enabled"
    }
  },
  "DHCP_SERVER_IPV4_CUSTOMIZED_OPTIONS":
  {
    "option60":
    {
      "id": 60,
      "type": "string",
      "value": "dummy_value"
    }
  },
  "DHCP_SERVER_IPV4_RANGE":
  { "range1":
    {
      "ip_start": "101.1.1.2",
      "ip_end": "101.1.1.6"
    }
  },
  "DHCP_SERVER_IPV4_PORT":
  {
    "Vlan100|PortChannel0003":{"ips": [ "101.1.1.10" ]},
    "Vlan100|PortChannel2":{"range":["range1" ]
  }
}
```



1.5.0.24 DSCP_TO_TC_MAP

```
{
  "DSCP_TO_TC_MAP":
    { "AZURE":
      {
        "1": "1",
        "0": "1",
        "3": "3",
        "2": "1",
        "5": "2",
        "4": "4",
        "7": "1",
        "6": "1",
        "9": "1",
        "8": "0"
      }
    }
}
```

1.5.0.25 FG_NHG

The FG_NHG table provides information on Next Hop Groups, including a specified Hash Bucket Size (bucket_size) and match mode for each group.

```
"FG_NHG":
{
  "fghhg_v4":
    {
      "bucket_size": "120",
      "match_mode": "nexthop-based"
    },
  "fghhg_v6":
    {
      "bucket_size": "120",
      "match_mode": "nexthop-based"
    }
}
```




1.5.0.26 FG_NHG_MEMBER

The FG_NHG_MEMBER table provides information about the members of a next hop group, including the group name (FG_NHG), the index at which redistribution is performed (bank), and the link associated with the next-hop-ip (link).

"FG_NHG_MEMBER":

```
{
  "100.100.100.1":
    {
      "FG_NHG": "fgnhg_v4",
      "bank": "0",
      "link": "Ethernet47" },
  "100.100.100.2":
    {
      "FG_NHG": "fgnhg_v4",
      "bank": "1",
      "link": "Ethernet48" }
}
```

1.5.0.27 FG_NHG_PREFIX

The FG_NHG_PREFIX table provides the FG_NHG_PREFIX for which FG behavior is desired, and Fine Grained next-hop group name.

"FG_NHG_PREFIX":

```
{
  "101.10.10.10/32": {"FG_NHG": "fgnhg_v4" },
  "fc:02::/128": {"FG_NHG": "fgnhg_v6" }
}
```

1.5.0.28 FABRIC_MONITOR

```
{
  "FABRIC_MONITOR":
    {
      "FABRIC_MONITOR_DATA":
        {
          "monErrThreshCrcCells": "1",
          "monErrThreshRxCells": "60023147",
          "monPollThreshIsolation": "1",
          "monPollThreshRecovery": "8",
          "monCapacityThreshWarn": "10",
          "monState": "enable"
        }
    }
}
```



1.5.0.29 FABRIC_PORT

```
{
  "FABRIC_PORT":
  {
    "Fabric0":
    {
      "alias": "Fabric0",
      "isolateStatus": "False",
      "lanes": "0",
      "forceUnisolateStatus": "0"
    },
    "Fabric1":
    {
      "alias": "Fabric1",
      "isolateStatus": "False",
      "lanes": "1",
      "forceUnisolateStatus": "0"
    }
  }
}
```

1.5.0.30 MPLS_TC_TO_TC_MAP

```
{
  "MPLS_TC_TO_TC_MAP":
  {
    "AZURE":
    {
      "0": "0",
      "1": "1",
      "2": "1",
      "3": "2",
      "4": "2",
      "5": "3",
      "6": "3",
      "7": "4"
    }
  }
}
```



1.5.0.31 FLEX_COUNTER_TABLE

```
{
  "FLEX_COUNTER_TABLE":
  {
    "PFCWD":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "10000"
    },
    "PORT":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "1000"
    },
    "QUEUE":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "10000"
    },
    "TUNNEL":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "10000"
    },
    "WRED_ECN_QUEUE":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "10000"
    },
    "WRED_ECN_PORT":
    {
      "FLEX_COUNTER_STATUS": "enable",
      "POLL_INTERVAL": "1000"
    }
  }
}
```



1.5.0.32 Hash

Generic hash allows user to configure various aspects of hashing algorithm. The configuration is applied globally for each ECMP and LAG on a switch.

ECMP/LAG HASH

```
{
  "SWITCH_HASH":
  {
    "GLOBAL":
    {
      "ecmp_hash":
      [
        "DST_MAC",
        "SRC_MAC",
        "ETHERTYPE",
        "IP_PROTOCOL",
        "DST_IP",
        "SRC_IP",
        "L4_DST_PORT",
        "L4_SRC_PORT",
        "INNER_DST_MAC",
        "INNER_SRC_MAC",
        "INNER_ETHERTYPE",
        "INNER_IP_PROTOCOL",
        "INNER_DST_IP",
        "INNER_SRC_IP",
        "INNER_L4_DST_PORT",
        "INNER_L4_SRC_PORT"
      ],
      "lag_hash":
      [
        "DST_MAC",
        "SRC_MAC",
        "ETHERTYPE",
        "IP_PROTOCOL",
        "DST_IP",
        "SRC_IP",
        "L4_DST_PORT",
        "L4_SRC_PORT",
        "INNER_DST_MAC",
        "INNER_SRC_MAC",
        "INNER_ETHERTYPE",
        "INNER_IP_PROTOCOL",
        "INNER_DST_IP",
        "INNER_SRC_IP",
        "INNER_L4_DST_PORT",
        "INNER_L4_SRC_PORT"
      ],
      "ecmp_hash_algorithm": "CRC",
      "lag_hash_algorithm": "CRC"
    }
  }
}
```



1.5.0.33 KDUMP

```
{
  "KDUMP":
  {
    "config":
    {
      "enabled": "true",
      "num_dumps": "3",
      "memory": "0M-2G:256M,2G-4G:256M,4G-8G:384M,8G-:448M",
      "remote": "true",
      "ssh_string": "username@ipaddress",
      "ssh_path": "a/b/c"
    }
  }
}
```

1.5.0.34 Kubernetes Master

Kubernetes Master related configurations are stored in **KUBERNETES_MASTER** table. These configurations are used mainly for CTRMGR service. CTRMGR service will interactive with kubernetes master according to these configurations.

Example,

```
{
  "KUBERNETES_MASTER":
  {
    "SERVER":
    {
      "disable": "False",
      "insecure": "True",
      "ip": "kubernetesDomain.com",
      "port": "6443"
    }
  }
}
```



1.5.0.35 L2 Neighbors

The L2 neighbor and connection information can be configured in **DEVICE_NEIGHBOR** table. Those information are used mainly for LLDP. While mandatory fields include neighbor name acting as object key and remote port / local port information in attributes, optional information about neighbor device such as device type, hwsku, management address and loopback address can also be defined.

```
{
  "DEVICE_NEIGHBOR":
  {
    "DQS5K04T1":
    {
      "mgmt_addr": "10.10.10.10",
      "hwsku": "D-LINK_DQS-5000-56ZS-A1",
      "lo_addr": null,
      "local_port": "Ethernet24",
      "type": "LeafRouter",
      "port": "Ethernet1"
    },
    "DQS5K03T1":
    {
      "mgmt_addr": "10.10.10.12",
      "hwsku": "D-LINK_DQS-5000-56ZS-A1",
      "lo_addr": null,
      "local_port": "Ethernet20",
      "type": "LeafRouter",
      "port": "Ethernet1"
    },
    "DQS5K02T1":
    {
      "mgmt_addr": "10.10.10.16",
      "hwsku": "D-LINK_DQS-5000-56ZS-A1",
      "lo_addr": null,
      "local_port": "Ethernet18",
      "type": "LeafRouter",
      "port": "Ethernet1"
    },
    "DQS5K01T1":
    {
      "mgmt_addr": "10.10.10.12",
      "hwsku": "D-LINK_DQS-5000-56ZS-A1",
      "lo_addr": null,
      "local_port": "Ethernet16",
      "type": "LeafRouter",
      "port": "Ethernet1"
    }
  }
}
```



1.5.0.36 L3 Interfaces

Configuration for L3 data plane interfaces are defined in the **INTERFACE**, **VLAN_INTERFACE**, **VLAN_SUB_INTERFACE** and **PORTCHANNEL_INTERFACE** tables, respectively.

The objects in all four tables have the interface as the key with the following required patterns:

- **INTERFACE** : Any valid port name from the **PORT** table. Typically **Ethernet{0-999}** .
- **VLAN_INTERFACE** : **Vlan{1-4095}**
- **PORTCHANNEL_INTERFACE** : **PortChannel{0-9999}**
- **VLAN_SUB_INTERFACE** : Any valid **INTERFACE** or **PORTCHANNEL_INTERFACE** name followed by a . and a number between 1 and 4094.

These tables have a number of shared attributes as described below:

- **vrf_name** : Valid VRF name from the **VRF** table. Default: **default**
- **nat_zone** : NAT Zone for this interface. **0..3**
- **mpls** : Enable/disable MPLS routing for the interface. **enable** or **disable** . Default **disable** .
- **ipv6_use_link_local_only** : Enable/Disable IPv6 link local address on interface. **enable** or **disable** . Default **disable** .
- **mac_addr** : Assign administrator-provided MAC address to Interface. If not specified will use the system MAC (same for all interfaces). Not applicable to **VLAN_SUB_INTERFACE** as it will use the parent interface's mac address.
- **loopback_action** : Packet action when a packet ingress and gets routed on the same IP interface. **drop** OR **forward** .

```
{
  "INTERFACE":
  {
    "Ethernet0":
    {
      "ipv6_use_link_local_only": "enable",
      "mac_addr": "12:34:56:78:90:ab"
    },
    "Ethernet1":
    {
      "loopback_action": "drop"
    }
  },
  "VLAN_INTERFACE":
  {
    "Vlan201":
    {
      "vrf_name": "red",
      "mac_addr": "12:34:56:ab:cd:12"
    }
  },
  "PORTCHANNEL_INTERFACE":
  {
    "PortChannel101":
    {
      "mac_addr": "12:34:56:bc:12:ab"
    }
  },
  "VLAN_SUB_INTERFACE":
  {
    "Ethernet0.200":
    {
      "vrf_name": "vrfName",
      "vlan": "200"
    }
  }
}
```



1.5.0.36.1 Assigning IP addresses to L3 Interfaces

The interface name and IP prefix act as multi-level key for the interface objects. The IP prefixes are stored in the same tables as the interface attributes: **INTERFACE** , **VLAN_INTERFACE** , **VLAN_SUB_INTERFACE** and **PORTCHANNEL_INTERFACE** .

In the example below we will show one interface with attributes for clarity, but otherwise this is simply an example of how IP addresses may be configured.

```
{
  "INTERFACE":
  {
    "Ethernet0":
    {
      "ipv6_use_link_local_only": "enable",
      "mac_addr": "12:34:56:78:90:ab"
    },
    "Ethernet0|10.10.10.0/31": {},
    "Ethernet4|10.10.10.2/31": {},
    "Ethernet8|10.10.10.4/31": {},
  },
  "PORTCHANNEL_INTERFACE":
  {
    "PortChannel01|10.10.10.56/31": {},
    "PortChannel01|FC02::71/126": {},
    "PortChannel02|10.10.10.57/31": {},
    "PortChannel02|FC02::72/126": {}
  },
  "VLAN_INTERFACE":
  { "Vlan1000|192.168.0.1/27": {} },
  "VLAN_SUB_INTERFACE":
  { "Ethernet4.1|10.10.10.2/31": {} }
}
```

1.5.0.37 Loopback Interface

Loopback interface configuration lies in **LOOPBACK_INTERFACE** table and has similar schema with data plane interfaces. The loopback device name and loopback IP prefix act as multi-level key for loopback interface objects. By default SONiC advertises Loopback interface IPv6 /128 subnet address as prefix with /64 subnet. To overcome this set “bgp_adv_lo_prefix_as_128” to true in **DEVICE_METADATA**

```
{
  "LOOPBACK_INTERFACE":
  {
    "Loopback0|10.10.10.32/32": {},
    "Loopback0|FC02:1::32/128": {}
  }
}
```




1.5.0.38 LOSSLESS_TRAFFIC_PATTERN

The LOSSLESS_TRAFFIC_PATTERN table stores parameters related to lossless traffic for dynamic buffer calculation

```
{
  "LOSSLESS_TRAFFIC_PATTERN":
  {
    "AZURE":
    {
      "mtu": "1024",
      "small_packet_percentage": "100"
    }
  }
}
```

1.5.0.39 Memory Statistics

The memory statistics configuration is stored in the **MEMORY_STATISTICS** table. This table is used by the memory statistics daemon to manage memory monitoring settings. The configuration allows enabling or disabling memory collection, specifying how frequently memory statistics are sampled, and defining how long the memory data is retained.

```
{
  "MEMORY_STATISTICS":
  { "memory_statistics": {
    "enabled": "false",
    "sampling_interval": "5",
    "retention_period": "15"
  }}
}
```

- **enabled**: Defines whether the memory statistics collection is active (true or false).
- **sampling_interval**: Interval between data collection.
- **retention_period**: Time to retain collected data.

1.5.0.40 Management Interface

Management interfaces are defined in **MGMT_INTERFACE** table. Object key is composed of management interface name and IP prefix. Attribute **gwaddr** specifies the gateway address of the prefix. **forced_mgmt_routes** attribute can be used to specify addresses / prefixes traffic to which are forced to go through management network instead of data network.

```
{ "MGMT_INTERFACE":
  {
    "eth0|10.11.11.11/16":
    {
      "gwaddr": "10.11.0.1"
    },
    "eth0|fc02:2::32/64":
    {
      "forced_mgmt_routes": [ "10.11.11.100/31", "10.25.0.100", "10.25.0.0/28" ],
      "gwaddr": "fc02:2::1"
    }
  }
}
```



1.5.0.41 Management port

```
{
  "MGMT_PORT":
    { "eth0": {
      "alias": "eth0",
      "admin_status": "up"
    }}
}
```

1.5.0.42 Management VRF

```
{ "MGMT_VRF_CONFIG":
  { "vrf_global":
    { "mgmtVrfEnabled": "true" }
  }
}
```

1.5.0.43 MAP_PFC_PRIORITY_TO_QUEUE

```
{ "MAP_PFC_PRIORITY_TO_QUEUE":
  { "AZURE":
    { "1": "1", "0": "0", "3": "3", "2": "2", "5": "5", "4": "4", "7": "7", "6": "6" }
  }
}
```

1.5.0.44 MUX_CABLE

The **MUX_CABLE** table is used for dualtor interface configuration. The `cable_type` and `soc_ipv4` objects are optional.

```
{ "MUX_CABLE":
  { "Ethernet4":
    {
      "cable_type": "active-active",
      "server_ipv4": "192.168.0.2/32",
      "server_ipv6": "fc02:100::30/128",
      "soc_ipv4": "192.168.0.3/32",
      "state": "auto"
    }
  }
}
```



1.5.0.45 MUX_LINKMGR

The **MUX_LINKMGR** table is used for dualtor device configuration.

```
{ "MUX_LINKMGR":
  { "LINK_PROBER":
    {
      "interval_v4": "100",
      "interval_v6": "1000",
      "positive_signal_count": "1",
      "negative_signal_count": "3",
      "suspend_timer": "500",
      "use_well_known_mac": "enabled",
      "src_mac": "ToRMac",
      "interval_pck_loss_count_update": "3"
    },
    "MUXLOGGER":
      { "log_verbosity": "debug"},
    "SERVICE_MGMT":
      { "kill_radv": "True" }
  }
}
```

1.5.0.46 NEIGH

The **NEIGH** table is used to keep track of resolved and static neighbors.

Resolve case:

```
{ "NEIGH":
  { "Vlan100|10.10.10.3":
    { "family": "IPv4" }
  }
}
```

Static Nbr:

```
{ "NEIGH":
  { "Vlan100|10.10.10.5":
    {
      "neigh": "01:23:45:67:89:a1",
      "family": "IPv4"
    }
  }
}
```



1.5.0.47 NTP Global Configuration

These configuration options are used to modify the way that ntp binds to the ports on the switch and which port it uses to make ntp update requests from.

NTP Admin state

If this option is set to `enabled` then ntp client will try to sync system time with configured NTP servers. Otherwise, NTP client feature will be disabled.

```
{
  "NTP": { "global": {
    "admin_state": "enabled" }
  }}
}
```

NTP Server role

This option is used to control NTP server state on the switch. If this option is set to `enabled` switch will act as NTP server. By default `server_role` is `disabled`.

```
{
  "NTP": { "global": {
    "server_role": "enabled" }
  }}
}
```

NTP VRF

If this option is set to `default` then ntp will run within the default vrf **when the management vrf is enabled**. If the mgmt vrf is enabled and this value is not set to default then ntp will run within the mgmt vrf.

This option **has no effect** if the mgmt vrf is not enabled.

```
{ "NTP":
  { "global":
    { "vrf": "default" }
  }
}
```

NTP Source Port

This option sets the port which ntp will choose to send time update requests from by.

NOTE: If a Loopback interface is defined on the switch ntp will choose this by default, so this setting is **required** if the switch has a Loopback interface and the ntp peer does not have defined routes for that address.

```
{ "NTP":
  { "global":
    { "src_intf": "Ethernet1" }
  }
}
```

NTP Authentication

If this option is set to `enabled` then ntp will try to verify NTP servers it connects to. This option **has no effect** if key is not set for NTP server. By default it is `disabled`.

```
{
  "NTP": { "global": {
    "authentication": "enabled" }
  }}
}
```



NTP Authentication

If this option is set to `enabled` then ntp will try to verify NTP servers it connects to. This option **has no effect** if key is not set for NTP server. By default it is `disabled`

```
{ "ntp":
  { "global":
    { "authentication": "enabled" }
  }
}
```

NTP DHCP leases

If this option is set to `enabled` then ntp client will try to use NTP servers provided by DHCP server. If this option is set to `disabled` you will be able to use the user-configured NTP servers. By default it is `enabled`

```
{ "ntp":
  { "global":
    { "dhcp": "enabled" }
  }
}
```

1.5.0.48 NTP servers

These information are configured in individual tables. Domain name or IP address of the server is used as object key. Currently there are no attributes in those objects.

NTP server

```
{
  "ntp_server":
  {
    "ntpserverDomain1": { "association_type": "pool", "iburst": "on", "admin_state": "enabled", "version": 4 },
    "ntpserverDomain2": { "association_type": "pool", "iburst": "off", "admin_state": "enabled", "version": 3 },
    "ntpserverDomain3": { "association_type": "pool", "iburst": "on", "admin_state": "disabled", "version": 4 },
    "ntpserverDomain4": { "association_type": "pool", "iburst": "off", "admin_state": "disabled", "version": 3 }
  },
  "ntp_server":
  {
    "xxx.xxx.xxx.xxx": { "association_type": "server", "iburst": "on", "admin_state": "enabled", "version": 4, "key": 3,
      "trusted": "yes"},
    "aaa.bbb.ccc.ddd": { "association_type": "server", "iburst": "on", "admin_state": "enabled", "version": 3 }
  }
}
```



- **association_type** - is used to control the type of the server. It can be `server` or `pool`.
- **iburst** - aggressive server polling `{on, off}`.
- **version** - NTP protocol version to use `[3..4]`.
- **key** - authentication key id `[1..65535]` to use to auth the server.
- **admin_state** - enable or disable specific server.
- **trusted** - trust this server when auth is enabled.

NTP keys

```
{
  "NTP_KEY":
    { "1": {
      "type": "md5",
      "value": "cYlwYXNzE31yAZ==",
      "trusted": "yes"
    }, "42": {
      "type": "sha1",
      "value": "CehlzW5zd2WX",
      "trusted": "no"
    }
  }
}
```

- **type** - key type to use `{md5, sha1, sha256, sha384, sha512}`.
- **value** - base64 encoded key value.
- **trusted** - trust this NTP key `{yes, no}`.

1.5.0.49 Peer Switch

Below is an example of the peer switch table configuration.

```
{ "PEER_SWITCH":
  { "dlab-05":
    { "address_ipv4": "10.10.10.1" }
  }
}
```

1.5.0.50 Policer

Below is an example of the policer table configuration.

```
{ "POLICER":
  { "everflow_static_policer":
    { "meter_type": "bytes",
      "mode": "sr_tcm",
      "cir": "12500000",
      "cbs": "12500000",
      "pir": "17500000",
      "pbs": "17500000",
      "color": "aware",
      "red_packet_action": "drop",
      "yellow_packet_action": "drop",
      "green_packet_action": "forward"
    }
  }
}
```

Key to the table defines policer name Below are the fields



- **meter_type** - Mandatory field. Defines how the metering is done. values - bytes, packets
- **mode** - Mandatory field. Defines one of the three modes support. values - sr_tcm, tr_tcm, storm
- **cir** - Committed information rate bytes/sec or packets/sec based on meter_type
- **cbs** - Committed burst size in bytes or packets based on meter_type
- **pir** - Peak information rate in bytes/sec or packets/sec based on meter_type
- **pbs** - Peak burst size in bytes or packets based on meter_type
- **color** - Defines the color source for the policer. values - aware, blind
- **red_packet_action** - Defines the action to be taken for red color packets
- **yellow_packet_action** - Defines the action to be taken for yellow color packets
- **green_packet_action** - Defines the action to be taken for green color packets.

The packet action could be:

- 'drop'
- 'forward'
- 'copy'
- 'copy_cancel'
- 'trap'
- 'log'
- 'deny'
- 'transit'

1.5.0.51 Port

In this table the physical port configurations are defined. Each object will have port name as its key, and port name alias and port speed as optional attributes.

```
{ "PORT":
{
  "Ethernet0":
  { "index": "0", "lanes": "101,102", "description": "fortyGigE1/1/1", "mtu": "9100", "alias": "fortyGigE1/1/1", "speed":
    "40000", "link_training": "off", "laser_freq": "191300", "tx_power": "-27.3", "dom_polling": "enabled", "core_id": "1",
    "core_port_id": "1", "num_voq": "8"},
  "Ethernet1":
  { "index": "1", "lanes": "103,104", "description": "fortyGigE1/1/2", "mtu": "9100", "alias": "fortyGigE1/1/2",
    "admin_status": "up", "speed": "40000", "link_training": "on", "laser_freq": "191300", "tx_power": "-27.3",
    "dom_polling": "enabled", "core_id": "0", "core_port_id": "14", "num_voq": "8"},
  "Ethernet63":
  { "index": "63", "lanes": "87,88", "description": "fortyGigE1/4/16", "mtu": "9100", "alias": "fortyGigE1/4/16", "speed":
    "40000", "laser_freq": "191300", "tx_power": "-27.3", "dom_polling": "disabled", "core_id": "0", "core_port_id": "15",
    "num_voq": "8"}
}
```

2x100G port breakout

```
{ "PORT":
{
  "Ethernet0":
  { "admin_status": "up", "index": "1", "lanes": "101,102,103,104", "description": "etp1a", "mtu": "9100", "alias":
    "etp1a", "speed": "100000", "subport": 1, "dom_polling": "enabled" },
  "Ethernet4":
  { "admin_status": "up", "index": "1", "lanes": "105,106,107,108", "description": "etp1b", "mtu": "9100", "alias":
    "etp1b", "speed": "100000", "subport": 2, "dom_polling": "enabled" },
}
```



1.5.0.52 Port Channel

Port channels are defined in **PORTCHANNEL** table with port channel name as object key and member list as attribute.

```
{ "PORTCHANNEL": { "PortChannel0003": {
    "admin_status": "up",
    "min_links": "1",
    "members": [ "Ethernet54" ],
    "mtu": "9100" },
  "PortChannel0004": {
    "admin_status": "up",
    "min_links": "1",
    "members": [ "Ethernet56" ],
    "mtu": "9100",
    "fallback": "false",
    "fast_rate": "true" }
  }
}
```

1.5.0.53 Portchannel member

```
{
  "PORTCHANNEL_MEMBER":
    { "PortChannel0001|Ethernet50": {},
      "PortChannel0002|Ethernet52": {},
      "PortChannel0003|Ethernet54": {},
      "PortChannel0004|Ethernet56": {}
    }
}
```

1.5.0.54 Scheduler

```
{
  "SCHEDULER": {
    "scheduler.0":
      {
        "type": "STRICT"
      },
    "scheduler.1":
      {
        "type": "WRR",
        "weight": "1",
        "meter_type": "bytes",
        "pir": "1250000000",
        "pbs": "8192"
      },
    "scheduler.port":
      {
        "meter_type": "bytes",
        "pir": "1000000000",
        "pbs": "8192"
      }
  }
}
```




1.5.0.55 Port QoS Map

```
{
  "PORT_QOS_MAP":
  {
    "Ethernet50,Ethernet52,Ethernet54,Ethernet56":
    {
      "tc_to_pg_map": "AZURE",
      "tc_to_queue_map": "AZURE",
      "pfc_enable": "3,4",
      "pfc_to_queue_map": "AZURE",
      "dscp_to_tc_map": "AZURE",
      "dscp_to_fc_map": "AZURE",
      "exp_to_fc_map": "AZURE",
      "scheduler": "scheduler.port"
    }
  }
}
```

In this example, AZURE tells SONiC to use the predefined QoS mapping profile provided in the system defaults.

1.5.0.56 Queue

```
{
  "QUEUE":
  {
    "Ethernet56|4":
    { "wred_profile": "AZURE_LOSSLESS", "scheduler": "scheduler.1" },
    "Ethernet56|5":
    { "scheduler": "scheduler.0" },
    "Ethernet56|6":
    { "scheduler": "scheduler.0" }
  }
}
```

1.5.0.57 Restapi

```
{
  "RESTAPI":
  { "certs": { "ca_cert": "/etc/sonic/credentials/ame_root.pem",
    "server_key": "/etc/sonic/credentials/restapiserver.key",
    "server_cert": "/etc/sonic/credentials/restapiserver.crt",
    "client_cert_cname": "client.sonic.net" },
    "config":
    { "client_auth": "true", "log_level": "trace", "allow_insecure": "false" }
  }
}
```



1.5.0.58 Sflow

The below are the tables and their schema for SFLOW feature

SFLOW

Field	Description	Mandatory	Default	Preference
admin_state	Global sflow admin state		down	
sample_direction	Global sflow sample direction		rx	
polling_interval	The interval within which sFlow data is collected and sent to the configured collectors		20	
agent_id	Interface name			PORT: name, PORTCHANNEL: name,MGMT VLAN: name

SFLOW_SESSION

Key - port

Field	Description	Mandatory	Default	Preference
Port	Sets sflow session table attributes for either all interfaces or a specific Eth-ernet interface.			PORT: name
admin_state	Per port sflow admin state		up	
sample_rate	Sets the packet sampling rate. The rate is expressed as an integer N, where the intended sampling rate is 1/N packets.			
sample_direction	Per port sflow sample direction		rx	
SFLOW_COLLECTOR				

key - name

Field	Description	Mandatory	Default	Preference
Name	Name of the Sflow collector			
collector_ip	IPv4/IPv6 address of the Sflow collector	true		
collector_port	Destination L4 port of the Sflow collector		6343	
collector_vrf	Specify the Collector VRF. In this revision, it is either default VRF or Management VRF.			

1.5.0.59 Storage Monitoring Daemon Interval Configuration

These options are used to configure the daemon polling and sync-to-disk interval of the Storage Monitoring Daemon (stormond)

Config Sample

```
{
  "STORMOND_CONFIG":
    { "INTERVALS": {
      "daemon_polling_interval" : "60",
      "fsstats_sync_interval"   : "360"
    }}
}
```

- **daemon_polling_interval** - Determines how often stormond queries the disk for relevant information and posts to STATE_DB
- **fsstats_sync_interval** - Determines how often key information from the STATE_DB is synced to a file on disk

1.5.0.60 Syslog Global Configuration

These configuration options are used to configure rsyslog utility and the way the system generates logs.

Configuration sample

```
{
  "SYSLOG_CONFIG":
  { "GLOBAL":
    {
      "rate_limit_interval": "5",
      "rate_limit_burst": "100",
      "format": "welf",
      "welf_firewall_name": "bla",
      "severity": "info"
    }
  }
}
```

- **rate_limit_interval** - determines the amount of time that is being measured for rate limiting: unsigned integer
- **rate_limit_burst** - defines the amount of messages, that have to occur in the time limit: unsigned integer
- **format** - syslog log format: {standard, welf}
- **welf_firewall_name** - WELF format firewall name: string
- **severity** – global log severity: {emerg, alert, crit, error, warning, notice, info, debug}

Syslog Rate Limit

Host side configuration:

```
{
  "SYSLOG_CONFIG":
  { "GLOBAL":
    { "rate_limit_interval": "300", "rate_limit_burst": "20000"
    }
  }
}
```

Container side configuration:

```
{
  "SYSLOG_CONFIG_FEATURE":
  {
    "bgp":
    {
      "rate_limit_interval": "300",
      "rate_limit_burst": "20000"
    },
    "pmon":
    {
      "rate_limit_interval": "300",
      "rate_limit_burst": "20000"
    }
  }
}
```



37

1.5.0.61 Syslog servers

These information are configured in individual tables. Domain name or IP address of the server is used as object key. Each server can be configurable.

Configuration sample

38



1.5.0.61 Syslog servers

These information are configured in individual tables. Domain name or IP address of the server is used as object key. Each server can be configurable.

Configuration sample

```
{
  "SYSLOG_SERVER":
  {
    "10.0.0.5": {},
    "10.0.0.6": {},
    "10.11.150.5": {}
  },
  "SYSLOG_SERVER" :
  {
    "4.4.4.4":
    {
      "source": "3.3.3.3",
      "port": "514",
      "vrf": "mgmt"
    },
    "2222::2222":
    {
      "source": "1111::1111",
      "port": "514",
      "vrf": "Vrf-Data"
    },
    "somehostname":
    {
      "filter": "include",
      "filter_regex": "ololo",
      "port": "514",
      "protocol": "tcp",
      "severity": "notice",
      "vrf": "default"
    }
  }
}
```

- **filter** - determines if syslog will include or exclude messages specified by regex: {include, exclude}
- **filter_regex** - filter messages by this regex: **string**
- **port** - network port to use to connect to remote server: integer: 1..65535
- **protocol** - network protocol to use to connect to remote server: {tcp, udp}
- **severity** - per-server log severity, overrides global one: {emerg, alert, crit, error, warning, notice, info, debug}

1.5.0.62 System Port

Every port on the system requires a global representation, known as a System Port, and is listed in this table.

```
{
  "SYSTEM_PORT": {
    "host227-4|asic0|Ethernet0": { "core_index": "1", "core_port_index": "1", "num_voq": "8",
      "speed": "100000", "switch_id": "0", "system_port_id": "1"
    },
    "host227-4|asic0|Ethernet4": { "core_index": "1",
```



1.5.0.62 System Port

Every port on the system requires a global representation, known as a System Port, and is listed in this table.

```
{
  "SYSTEM_PORT":
  {
    "host227-4|asic0|Ethernet0":
    {
      "core_index": "1",
      "core_port_index": "1",
      "num_voq": "8",
      "speed": "100000",
      "switch_id": "0",
      "system_port_id": "1"
    },
    "host227-4|asic0|Ethernet4":
    {
      "core_index": "1",
      "core_port_index": "2",
      "num_voq": "8",
      "speed": "100000",
      "switch_id": "0",
      "system_port_id": "2"
    },
    "host227-5|asic0|Ethernet0":
    {
      "core_index": "1",
      "core_port_index": "1",
      "num_voq": "8",
      "speed": "100000",
      "switch_id": "4",
      "system_port_id": "80"
    },
    "host227-5|asic0|Ethernet4":
    {
      "core_index": "1",
      "core_port_index": "2",
      "num_voq": "8",
      "speed": "100000",
      "switch_id": "4",
      "system_port_id": "81"
    }
  }
}
```



1.5.0.63 Tacplus Server

```
{
  "TACPLUS_SERVER":
    {
      "10.0.0.8":
        {
          "priority": "1",
          "tcp_port": "49"
        },
      "10.0.0.9":
        {
          "priority": "1",
          "tcp_port": "49"
        }
    }
}
```

1.5.0.64 TC to Priority group map

```
{
  "TC_TO_PRIORITY_GROUP_MAP":
    {
      "AZURE":
        {
          "1": "1", "0":
            "0", "3": "3",
            "2": "2", "5":
              "5", "4": "4",
              "7": "7", "6":
                "6"
          }
        }
}
```

1.5.0.65 TC to Queue map

```
{
  "TC_TO_QUEUE_MAP":
    {
      "AZURE":
        {
          "1": "1", "0":
            "0", "3": "3",
            "2": "2", "5":
              "5", "4": "4",
              "7": "7", "6":
                "6"
          }
        }
}
```



1.5.0.66 Telemetry

```
{
  "TELEMETRY":
    {
      "certs":
        {
          "ca_cert": "/etc/sonic/telemetry/dsmsroot.cer",
          "server_cert": "/etc/sonic/telemetry/streamingtelemetryserver.cer",
          "server_key": "/etc/sonic/telemetry/streamingtelemetryserver.key"
        },
      "gnmi":
        {
          "client_auth": "true",
          "log_level": "2",
          "port": "50051",
          "save_on_set": "false"
        }
    }
}
```

1.5.0.67 Telemetry client

```
{
  "TELEMETRY_CLIENT":
    {
      "Global":
        {
          "encoding": "JSON_IETF",
          "retry_interval": "30",
          "src_ip": "30.57.185.38",
          "unidirectional": "true"
        },
      "DestinationGroup|HS":
        {
          "dst_addr": "30.57.186.214:8081,30.57.185.39:8081"
        },
      "Subscription|HS_RDMA":
        {
          "dst_group": "HS",
          "path_target": "COUNTERS_DB",
          "paths": "COUNTERS/Ethernet*,COUNTERS_PORT_NAME_MAP",
          "report_interval": "5000",
          "report_type": "periodic"
        }
    }
}
```




1.5.0.68 Tunnel

This table configures the MUX tunnel for Dual-ToR setup

```
{
  "TUNNEL":
  {
    "MuxTunnel0":
    {
      "dscp_mode": "uniform",
      "dst_ip": "10.1.0.32",
      "ecn_mode": "copy_from_outer",
      "encap_ecn_mode": "standard",
      "ttl_mode": "pipe",
      "tunnel_type": "IPINIP"
    }
  }
}
```

different example for configuring MUX tunnel

```
{
  "TUNNEL":
  {
    "MuxTunnel0":
    {
      "dscp_mode": "pipe",
      "dst_ip": "10.1.0.32",
      "ecn_mode": "standard",
      "encap_ecn_mode": "standard",
      "ttl_mode": "uniform",
      "tunnel_type": "IPINIP"
    }
  }
}
```

example mux tunnel configuration for when tunnel_qos_remap is enabled

```
{
  "TUNNEL":
  { "MuxTunnel0": {
    "tunnel_type": "IPINIP",
    "src_ip": "10.1.0.33",
    "dst_ip": "10.1.0.32",
    "dscp_mode": "pipe",
    "encap_ecn_mode": "standard",
    "ecn_mode": "copy_from_outer",
    "ttl_mode": "uniform",
    "decap_dscp_to_tc_map": "DecapDscpToTcMap",
    "decap_tc_to_pg_map": "DecapTcToPgMap",
    "encap_tc_to_dscp_map": "EncapTcToQueueMap",
    "encap_tc_to_queue_map": "EncapTcToDscpMap" }
  }
}
```



1.5.0.69 Versions

This table is where the current version of the software is recorded.

```
{
  "VERSIONS":
  {
    "DATABASE":
    {
      "VERSION": "version_1_0_1"
    }
  }
}
```

1.5.0.70 VLAN

This table is where VLANs are defined. VLAN name is used as object key, and member list as well as an integer id are defined as attributes. If a DHCP relay is required for this VLAN, a `dhcp_servers` attribute must be specified for that VLAN, the value of which is a list that must contain the domain name or IP address of one or more DHCP servers.

```
{
  "VLAN":
  {
    "Vlan1000":
    {
      "dhcp_servers": [ "192.0.0.1", "192.0.0.2", "192.0.0.3", "192.0.0.4" ],
      "members": [ "Ethernet0", "Ethernet4", "Ethernet8", "Ethernet12" ],
      "vlanid": "1000" }
  }
}
```

1.5.0.71 VLAN_MEMBER

VLAN member table has Vlan name together with physical port or port channel name as object key, and tagging mode as attributes.

```
{
  "VLAN_MEMBER":
  {
    "Vlan1000|PortChannel47":
    { "tagging_mode": "untagged" },
    "Vlan1000|Ethernet8":
    { "tagging_mode": "untagged" },
    "Vlan2000|PortChannel47":
    { "tagging_mode": "tagged" }
  }
}
```



1.5.0.72 VNET

VNET table has Vnet name as the object key, and vxlan_tunnel name, scope, vni, peer list, advertised prefix, src mac, and overlay dest mac as attributes. The vxlan_tunnel name (mandatory) is the tunnel name from the VXLAN table. scope (optional) must "default", vni (mandatory) is the vxlan tunnel vni, peer_list (optional) is for Vnet peering, advertise_prefix (optional) is used to allow advertisement of this vnet's routes, overlay_dmac (optional) is the mac address which is used for VNET ping monitoring sessions for the vnet routes and is optional.

```
{
  "VNET":
  {
    "Vnet1-1":
    {
      "vxlan_tunnel": "vtep1",
      "scope": "default",
      "vni": "10011",
      "peer_list": "",
      "advertise_prefix": "true",
      "overlay_dmac": "22:33:44:55:66:77"
    },
    "Vnetv4_v4-0":
    {
      "vxlan_tunnel": "vtep2",
      "scope": "default",
      "vni": "10011",
    }
  }
}
```

1.5.0.73 VNET_ROUTE_TUNNEL

VNET_ROUTE_TUNNEL table has vnet_name|prefix as the object key, where vnet_name is the name of the VNet and prefix is the ip4 prefix associated with the route tunnel. The table includes the following attributes:

- **ENDPOINT:** The endpoint/nexthop tunnel IP (mandatory). It is used to identify the endpoint of the tunnel.
- **MAC_ADDRESS:** The inner destination MAC address in the encapsulated packet (optional). It should be a 12-hexadeimal digit value.
- **VNI:** The VNI value in the encapsulated packet (optional). It should be a numeric value.

```
{
  "VNET_ROUTE_TUNNEL":
  {
    "Vnet_2000|100.100.1.1/32": {"endpoint": "192.168.1.1", "mac_address": "f9:22:83:99:22:a2"},
    "Vnetv4_v4-0|10.0.1.0/24": {"endpoint": "192.168.1.2", "mac_address": "f8:22:83:99:22:a2", "vni": "10012"}
  }
}
```



1.5.0.74 VOQ INBAND INTERFACE

VOQ_INBAND_INTERFACE holds the name of the inband system port dedicated for cpu communication. At this time, only inband_type of “port” is supported

"VOQ_INBAND_INTERFACE":

```
{
  "Ethernet-IB0": { "inband_type": "port" },
  "Ethernet-IB0|3.3.3.1/32": {},
  "Ethernet-IB0|3333::3:5/128": {}
}
```

1.5.0.75 VXLAN

VXLAN_TUNNEL holds the VTEP source ip configuration. VXLAN_TUNNEL_MAP holds the vlan to vni and vni to vlan mapping configuration. VXLAN_EVPN_NVO holds the VXLAN_TUNNEL object to be used for BGP-EVPN discovered tunnels.

```
{
  "VXLAN_TUNNEL":
    {"vtep1": { "src_ip": "10.10.10.10", "dst_ip": "12.12.12.12" }}
  "VXLAN_TUNNEL_MAP" :
    {"vtep1|map_1000_Vlan100": {"vni": "1000", "vlan": "100"},
     "vtep1|testmap": {"vni": "22000", "vlan": "70"},}
  "VXLAN_EVPN_NVO":
    { "nvo1": { "source_vtep": "vtep1" }
    }
}
```

1.5.0.76 Virtual router

The virtual router table allows to insert or update a new virtual router instance. The key of the instance is its name. The attributes in the table allow to change properties of a virtual router. Attributes:

- ‘v4’ contains boolean value ‘true’ or ‘false’. Enable or disable IPv4 in the virtual router
- ‘v6’ contains boolean value ‘true’ or ‘false’. Enable or disable IPv6 in the virtual router
- ‘src_mac’ contains MAC address. What source MAC address will be used for packets egressing from the virtual router
- ‘ttl_action’ contains packet action. Defines the action for packets with TTL == 0 or TTL ==1
- ‘ip_opt_action’ contains packet action. Defines the action for packets with IP options
- ‘l3_mc_action’ contains packet action. Defines the action for unknown L3 multicast packets
- ‘vni’ contains L3 VNI value. VNI associated Virtual router instance.

The packet action could be:

- ‘drop’
- ‘forward’
- ‘copy’
- ‘copy_cancel’
- ‘trap’
- ‘log’
- ‘deny’
- ‘transit’

```
'VRF:rid1':
{ 'v4': 'true', 'v6': 'false', 'src_mac': '02:04:05:06:07:08',
  'ttl_action': 'copy', 'ip_opt_action': 'deny', 'l3_mc_action': 'drop', 'vni': '100'
}
```



1.5.0.77 WRED_PROFILE

```
{
  "WRED_PROFILE":
    { "AZURE_LOSSLESS": {
      "red_max_threshold": "2097152",
      "wred_green_enable": "true",
      "ecn": "ecn_all",
      "green_min_threshold": "1048576",
      "red_min_threshold": "1048576",
      "wred_yellow_enable": "true",
      "yellow_min_threshold": "1048576",
      "green_max_threshold": "2097152",
      "green_drop_probability": "5",
      "yellow_max_threshold": "2097152",
      "wred_red_enable": "true",
      "yellow_drop_probability": "5",
      "red_drop_probability": "5"
    }
  }
}
```

1.5.0.78 Logger

In this table, the loglevel and logoutput of the components are defined. Each component will have the component name as its key; and LOGLEVEL and LOGOUTPUT as attributes. The LOGLEVEL attribute will define the verbosity of the component. The LOGOUTPUT attribute will define the file of printing the logs.

```
{
  "LOGGER":
    { "orchagent":
      { "LOGLEVEL": "NOTICE",
        "LOGOUTPUT": "SYSLOG"},
      "syncd":
        { "LOGLEVEL": "DEBUG", "LOGOUTPUT": "STDOUT"},
      "SAI_API_LAG":
        { "LOGLEVEL": "ERROR", "LOGOUTPUT": "STDERR"}
    }
}
```



1.5.0.79 PASSWORD_HARDENING

Password Hardening, a user password is the key credential used in order to verify the user accessing the switch and acts as the first line of defense in regards to securing the switch. PASSWORD_HARDENING - support the enforce strong policies.

- **state** - Enable/Disable password hardening feature
- **len_min** - The minimum length of the PW should be subject to a user change.
- **expiration** - PW Age Change Once a PW change takes place - the DB record for said PW is updated with the new PW value and a fresh new age (=0).
- **expiration_warning** - The switch will provide a warning for PW change before and (this is to allow a sufficient warning for upgrading the PW which might be relevant to numerous switches).
- **history_cnt** - remember last passwords, and reject to use the old passw
- **reject_user_passw_match** - reject to set same username and passw
- **PW classes** - are the type of characters the user is required to enter when setting/updating a PW.

There are 4 classes

- **lower_class** - Small characters - a-z
- **upper_class** - Big characters - A-Z
- **digits_class** - Numbers - 0-9
- **special_class** - **Special Characters** `~!@#\$%^&*()-_+=|[{]};, '<.>/? and white space

```
{
  "PASSW_HARDENING":
    { "POLICIES":
      {
        "state": "disabled",
        "expiration": "180",
        "expiration_warning": "15",
        "history_cnt": "10",
        "len_min": "8",
        "reject_user_passw_match": "true",
        "lower_class": "true",
        "upper_class": "true",
        "digits_class": "true",
        "special_class": "true"
      }
    }
}
```



1.5.0.80 SERIAL_CONSOLE

In this table collected configuration of the next serial-console attributes:

- **inactivity_timeout** - Inactivity timeout for serial-console session, allowed values: 0-35000 (minutes), default value: 15
- **sysrq_capabilities** - Enabling or disabling SysRq functionality for serial-console session, allowed values: enabled/disabled, default value disabled

```
{
  SERIAL_CONSOLE:
    {
      "POLICIES":
        {"inactivity_timeout": 15 "sysrq_capabilities": "disabled"}
    }
}
```

1.5.0.81 SSH_SERVER

In this table, we allow configuring ssh server global settings. This will feature includes 5 configurations:

- **authentication_retries** - number of login attempts 1-100
- **login_timeout** - Timeout in seconds for login session for user to connect 1-600
- **ports** - Ssh port numbers - string of port numbers separated by ‘;
- **inactivity_timeout** - Inactivity timeout for SSH session, allowed values: 0-35000 (min), default value: 15 (min)
- **max_sessions** - Max number of concurrent logins, allowed values: 0-100 (where 0 means no limit), default value: 0

```
{
  "SSH_SERVER":
    {
      "POLICIES":
        {
          "authentication_retries": "6",
          "login_timeout": "120",
          "ports": "22",
          "inactivity_timeout": "15",
          "max_sessions": "0"
        }
    }
}
```

1.5.0.82 BREAKOUT_CFG

This table is introduced as part of Dynamic Port Breakout(DPB) feature. It shows the current breakout mode of all ports(root ports). The list of root ports, all possible breakout modes, and default breakout modes are obtained/derived from platform.json and hwsku.json files.

```
"BREAKOUT_CFG":
{
  "Ethernet0": {"brkout_mode": "4x25G[10G]"},
  "Ethernet4": {"brkout_mode": "4x25G[10G]"},
  "Ethernet8": {"brkout_mode": "4x25G[10G]"},
  .....
  "Ethernet116": {"brkout_mode": "2x50G"},
  "Ethernet120": {"brkout_mode": "2x50G"},
  "Ethernet124": {"brkout_mode": "2x50G"}
}
```



1.5.0.83 AAA

The AAA table defined the method SONiC used for Authentication, Authorization and Accounting. The method could be:

- **default**
- **local**
- **tacacs+**
- **radius**

```
"AAA":
{
  "authentication":{"login": "local" },
  "authorization": { "login": "local"},
  "accounting": { "login": "local" }
}
```

1.5.0.84 SUBNET_DECAP

The **SUBNET_DECAP** table is used for subnet decap configuration.

```
"SUBNET_DECAP":
{ "AZURE": { "status": "enable", "src_ip": "10.10.10.0/24", "src_ip_v6": "20c1:ba8::/64" } }
```

1.5.0.85 SYSTEM_DEFAULTS table

To have a better management of the features in SONiC, a new table **SYSTEM_DEFAULTS** is introduced.

```
"SYSTEM_DEFAULTS":
{
  "tunnel_qos_remap": { "status": "enabled" }
  "default_bgp_status": { "status": "down"}
  "synchronous_mode": { "status": "enable"}
  "dhcp_server": { "status": "enable"}
}
```

The default value of flags in **SYSTEM_DEFAULTS** table can be set in **init_cfg.json** and loaded into db at system startup. These flags are usually set at image being build, and are unlikely to change at runtime.

If the values in **config_db.json** is changed by user, it will not be rewritten back by **init_cfg.json** as **config_db.json** is loaded after **init_cfg.json** in **docker_image_ctl.j2**

For the flags that can be changed by reconfiguration, we can update entries in **minigraph.xml** , and parse the new values in to **config_db** with minigraph parser at reloading minigraph. If there are duplicated entries in **init_cfg.json** and **minigraph.xml** , the values in **minigraph.xml** will overwritten the values defined in **init_cfg.json** .

1.5.0.86 RADIUS

The RADIUS and RADIUS_SERVER tables define RADIUS configuration parameters. RADIUS table carries global configuration while RADIUS_SERVER table carries per server configuration.

```
"RADIUS":
{ "global": { "auth_type": "pap", "timeout": "5" } }
"RADIUS_SERVER":
{ "192.168.1.2":
  { "priority": "4", "retransmit": "2", "timeout": "5" } }
```




1.5.0.87 Static DNS

The `DNS_NAMESERVER` table introduces static DNS nameservers configuration.

```
{
  "DNS_NAMESERVER":
  {
    "1.1.1.1": {},
    "fe80:1000:2000:3000::1": {}
  },
}
```

1.5.0.88 SRv6

The `SRV6_MY_SIDS` and `SRV6_MY_LOCATORS` tables introduce Segment Routing over IPv6 configuration. An example is as follows:

```
{
  "SRV6_MY_LOCATORS" :
  { "loc1" :
    { "prefix" : "FCBB:BBBB:20::" }
  }
  "SRV6_MY_SIDS" :
  {
    "loc1|FCBB:BBBB:20::" : { "action": "uN" },
    "loc1|FCBB:BBBB:20:F1::" : { "action": "uDT46", "decap_vrf": "default", "decap_dscp_mode": "pipe" }
  }
}
```

1.5.0.89 Prefix List

Prefix list table stores a list of prefixes with type and prefix separated by `|`. The specific configuration for the prefix type are then rendered by the PrefixListMgr. Currently `ANCHOR_PREFIX` is supported to add RADIANT configuration.

An example is as follows:

```
{
  "PREFIX_LIST":
  {
    "ANCHOR_PREFIX|fc00::/48": {}
  }
}
```

1.5.0.90 FIPS

The `FIPS` table introduces FIPS configuration.

```
{
  "FIPS":
  { "global" :
    { "enable": "true", "enforce": "false" }
  }
}
```



1.5.0.91 MID_PLANE_BRIDGE

The MID_PLANE_BRIDGE table introduces the configuration for the midplane bridge interface for Smart Switch.

```
{
  "MID_PLANE_BRIDGE":
    { "GLOBAL" : { "bridge": "bridge-midplane", "ip_prefix": "169.254.200.254/24" } }
}
```

1.5.0.92 DPUS

The DPUS table introduces the information on the DPUs (Data Processing Unit) available on the platform.

```
{
  "DPUS":
    {
      "dpu0": {"midplane_interface": "dpu0" },
      "dpu1": {"midplane_interface": "dpu1" }
    }
}
```

1.5.0.93 ASIC_SENSORS

The ASIC_SENSORS table introduces the asic sensors polling configuration when they are available on the platform.

```
{
  "ASIC_SENSORS":
    {
      "ASIC_SENSORS_POLLER_INTERVAL": {"interval": "10" },
      "ASIC_SENSORS_POLLER_STATUS": { "admin_status": "enable" }
    }
}
```



1.5.0.94 DPU PORT Configuration^M

The **DPU_PORT** table introduces the configuration for the DPUs(Data Processing Unit) PORT information available on the platform.

```
{
  "DPU_PORT":
  {
    "dpu0":
    {
      "state": "up",
      "vip_ipv4": "192.168.1.1",
      "vip_ipv6": "2001:db8::10",
      "pa_ipv4": "192.168.1.10",
      "pa_ipv6": "2001:db8::10",
      "vdpu_id": "vdpu0",
      "gnmi_port": "50052"
    },
    "dpu1":
    {
      "state": "down",
      "vip_ipv4": "192.168.1.2",
      "vip_ipv6": "2001:db8::20",
      "pa_ipv4": "192.168.1.20",
      "pa_ipv6": "2001:db8::20",
      "vdpu_id": "vdpu1",
      "gnmi_port": "50052"
    }
  }
}
```

state: Administrative status of the DPU (up or down).

vip_ipv4: VIP IPv4 address from minigraph.

vip_ipv6: VIP IPv6 address from minigraph.

pa_ipv4: PA IPv4 address from minigraph.

pa_ipv6: PA IPv6 address from minigraph.

vdpu_id: ID of VDPUs from minigraph.

gnmi_port: Port gNMI runs on.



1.6 For Developers

1.6.1 Generating Application Config by Jinja2 Template

To be added.

1.6.2 Incremental Configuration by Subscribing to ConfigDB

Detail instruction to be added. A sample could be found in this PR that implemented dynamic configuration for BGP.

2 Command reference

2.1 SONiC Command Line Interface Guide

2.1.1 Introduction

SONiC is an open source network operating system based on Linux that runs on switches from multiple vendors and ASICs. SONiC offers a full-suite of network functionality, like BGP and RDMA, that has been production-hardened in the data centers of some of the largest cloud-service providers. It offers teams the flexibility to create the network solutions they need while leveraging the collective strength of a large ecosystem and community.

SONiC software shall be loaded in these supported devices and this CLI guide shall be used to configure the devices as well as to display the configuration, state and status.

Follow the [Quick Start Guide](#) to boot the device in ONIE mode, install the SONiC software using the steps specified in the document and login to the device using the default username and password.

After logging into the device, SONiC software can be configured in following three methods.

1. **Command Line Interface (CLI)**
2. **config_db.json**
3. **minigraph.xml**

This document explains the first method and gives the complete list of commands that are supported in SONiC 201904 version (build#19). All the configuration commands need root privileges to execute them. Note that show commands can be executed by all users without the root privileges. Root privileges can be obtained either by using “sudo” keyword in front of all config commands, or by going to root prompt using “sudo -i”. Note that all commands are case sensitive.

- **Example:**

```
admin@sonic:~$ sudo config aaa authentication login tacacs+
```

OR

```
admin@sonic:~$ sudo -i
```

```
root@sonic:~# config aaa authentication login tacacs+
```

Note that the command list given in this document is just a subset of all possible configurations in SONiC. Please follow config_db.json based configuration for the complete list of configuration options.



Scope of this Document

It is assumed that all configuration commands start with the keyword “**config**” as prefix. Any other scripts/utilities/commands that need user configuration control are wrapped as sub-commands under the “config” command. The direct scripts/utilities/commands (examples given below) that are not wrapped under the “config” command are not in the scope of this document.

1. **acl_loader** – This script is already wrapped inside “**config acl**” command; i.e. any ACL configuration that user is allowed to do is already part of “config acl” command; users are not expected to use the **acl_loader** script directly and hence this document need not explain the “**acl_loader**” script.
2. **crm** – this command is not explained in this document.
3. sonic-clear, sfputil, etc., This document does not explain these scripts also.

2.1.2 Basic Tasks

This section covers the basic configurations related to the following:

1. **SSH login**
2. **Configuring the Management Interface**

2.1.2.1 SSH Login

All SONiC devices support both the serial console based login and the SSH based login by default. The default credential (if not modified at image build time) for login is **admin/YourPaSsWoRd**. In case of SSH login, users can login to the management interface (eth0) IP address after configuring the same using serial console. Refer the following section for configuring the IP address for management interface.

- Example:

At Console:

```
Debian GNU/Linux 9 sonic ttyS1
```

```
sonic login: admin Password:
```

```
YourPaSsWoRd
```

```
SSH from any remote server to sonic can be done by connecting to SONiC IP user@debug:
```

```
~$ ssh admin@sonic_ip_address(or SONIC DNS Name)
```

```
admin@sonic's password:
```

By default, login takes the user to the default prompt from which all the show commands can be executed.

Go Back To Beginning of the document or Beginning of this section



2.1.2.2 Show Management Interface

Please check show ip interfaces

2.1.2.3 Configuring Management Interface

The management interface (eth0) in SONiC is configured (by default) to use DHCP client to get the IP address from the DHCP server. Connect the management interface to the same network in which your DHCP server is connected and get the IP address from DHCP server. The IP address received from DHCP server can be verified using the `/sbin/ifconfig eth0` Linux command.

SONiC provides a CLI to configure the static IP for the management interface. There are few ways by which a static IP address can be configured for the management interface.

1. Use the `config interface ip add eth0` command. •

Example:

```
admin@sonic:~$ sudo config interface ip add eth0 20.11.12.13/24 20.11.12.254
```

2. Use `config_db.json` and configure the `MGMT_INTERFACE` key with the appropriate values. Refer here
3. Use `minigraph.xml` and configure “`ManagementIPInterfaces`” tag inside “`DpgDesc`” tag as given at the page

Once the IP address is configured, the same can be verified using either `show management_interface address` command or the `/sbin/ifconfig eth0` linux command.

Users can SSH login to this management interface IP address from their management network.

- Example:

```
admin@sonic:~$ /sbin/ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.11.11.13  netmask 255.255.255.0  broadcast 10.11.12.255
```

Go Back To Beginning of the document or Beginning of this section

2.1.3 Getting Help

Subsections:

1. Help for Config Commands
2. Help for Show Commands
3. Show Versions
4. Show System Status
5. Show Hardware Platform



2.1.3.1 Help for Config Commands

All commands have in-built help that aids the user in understanding the command as well as the possible sub-commands and options. “-help” can be used at any level of the command; i.e. it can be used at the command level, or sub-command level or at argument level. The in-built help will display the available possibilities corresponding to that particular command/sub-command.

config -help

This command lists all the possible configuration commands at the top

level. • Usage:

```
config --help
```

• Example:

```
admin@sonic:~$ config --help
Usage: config [OPTIONS] COMMAND [ARGS] SONiC
command line - 'config' command

Options:
  --help  Show this message and exit.

Commands:
  aaa                AAA command line
  acl                ACL-related configuration tasks BGP-
  bgp                related configuration tasks ECN-related
  ecn                configuration tasks Modify configuration of
  feature            features
  hostname            Change device hostname without impacting traffic Interface-
  interface           related configuration tasks
  interface_naming_mode Modify interface naming mode for interacting... Kubernetes
  kubernetes          server related configuration
  load                Import a previous saved config DB dump file. Reconfigure
  load_mgmt_config    hostname and mgmt interface based... Reconfigure based on
  load_minigraph       minigraph.
  loopback            Loopback-related configuration tasks.
  mirror_session nat
  platform            NAT-related configuration tasks Platform-related
  portchannel qos     configuration tasks
  reload
  route
  save                Clear current configuration and import a... route-related
  tacacs              configuration tasks
  vlan vrf            Export current config DB to a file on disk. TACACS+ server
  watermark            configuration
  warm_restart        VLAN-related configuration tasks VRF-
  watermark            related configuration tasks
  watermark            warm_restart-related configuration tasks
  watermark            Configure watermark
```

[Go Back To Beginning of the document or Beginning of this section](#)



2.1.3.2 Help For Show Commands

show help

This command displays the full list of show commands available in the software; the output of each of those show commands can be used to analyze, debug or troubleshoot the network node.

- Usage:

```
show (-?|-h|--help)
```

- Example:

```
admin@sonic:~$ show -?
```

```
Usage: show [OPTIONS] COMMAND [ARGS]... SONiC
```

```
command line - 'show' command
```

```
Options:
```

```
-?, -h, --help Show this message and exit.
```

```
Commands:
```

aaa	Show AAA configuration
acl	Show ACL related information Show IP
arp	ARP table
buffer_pool	Show details of the Buffer-pools Show date
clock	and time
ecn	Show ECN configuration
environment	Show environmental (voltages, fans, temps)
feature	Show feature status
interfaces ip	Show details of the network interfaces Show IP (IPv4) commands
ipv6	Show IPv6 commands
kubernetes	Show kubernetes commands
line	Show all /dev/ttyUSB lines and their info Show LLDP
lldp	information
logging	Show system log
mac	Show MAC (FDB) entries
mirror_session mmu	Show existing everflow sessions Show mmu configuration
muxcable	Show muxcable information
nat	Show details of the nat
ndp	Show IPv6 Neighbour table
ntp	Show NTP information
pfc	Show details of the priority-flow-control...
platform priority-	Show platform-specific hardware info
group processes	Show details of the PGs Show process information
queue	Show details of the queues
reboot-cause	Show cause of most recent reboot
route-map	Show route-map
runningconfiguration	Show current running configuration...
services	Show all daemon services
startupconfiguration	Show startup configuration information
subinterfaces	Show details of the sub port interfaces
system-memory	Show memory information
tacacs	Show TACACS+ configuration
techsupport	Gather information for troubleshooting



uptime	Show system uptime
users	Show users
version	Show version information
vlan	Show VLAN information
vrf	Show vrf config
warm_restart	Show warm restart configuration and state
watermark	Show details of watermark

The same syntax applies to all subgroups of `show` which themselves contain subcommands, and subcommands which accept options/arguments.

- Example:

```
admin@sonic:~$ show interfaces -?
```

Show details of the network interfaces

Options:

-, -h, --help Show this message and exit.

Commands:

counters	Show interface counters
description fec	Show interface status, protocol and...Show interface fec information
link-training	Show interface link-training information
naming_mode	Show interface naming_mode status
neighbor	Show neighbor related information
portchannel	Show PortChannel information
status	Show Interface status information
tpid	Show Interface tpid information
transceiver	Show SFP Transceiver information

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.4 Basic Show Commands

Subsections:

1. Show Versions
2. Show System Status
3. Show Hardware Platform

2.1.4.1 Show Versions

show version

This command displays software component versions of the currently running SONiC image. This includes the SONiC image version as well as Docker image versions. This command displays relevant information as the SONiC and Linux kernel version being utilized, as well as the ID of the commit used to build the SONiC image. The second section of the output displays the various docker images and their associated IDs.

- Usage:

```
show version
```

- Example:

```
admin@sonic:~$ show version
SONiC Software Version: SONiC OS ver.12_DQS-5000-56ZS
SONiC OS Version: 12
Distribution: Debian 12.11
Build commit: Thu Jul 31 17:49:08 UTC 2025
Build by: gitlab-runner@node05
Platform: x86_64- 5000-56ZS-A1
HwSKU: D-Link_DQS-5000-56ZS-A1
ASIC: marvell
ASIC Count: 1
Serial Number:
Model Number: DQS-5000056ZS
Hardware Rev: A1
Uptime: 11:27:17 up 1:35, 1 user, load average: 0.96, 1.05, 1.18
Date: Fri 15 Mar 2025 11:27:17
```

Docker images:

REPOSITORY	TAG	IMAGE ID	SIZE
docker-l2mcd	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	eb9f625effbb	361MB
docker-syncd-mrvl	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	9a237c7181df	620MB
docker-orchagent	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	fabf618f261e	356MB
docker-macsec	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	7cc4c72d5ccc	346MB
docker-teamd	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	0d1a20d5d68d	343MB
docker-sflow	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	36b7424a0b83	344MB
docker-fpm-frr	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	5d96c4b64557	377MB
docker-nat	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	10f1c16f4f32	346MB
docker-iccpd	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	3b142c17acd8	346MB
docker-stp	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	3decbb4cef91	361MB
Docker-lldp	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	86ff37817d10	358MB
docker-sonic-bmp	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	1ac6d944a77b	313MB
docker-eventd	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	165cee7fe808	312MB
docker-dhcp-relay	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	31cb4d799053	321MB
docker-dhcp-server	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	15764d3b0c1d	335MB
docker-snmp	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	c471ec284670	356MB
docker-platform-monitor	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	00379f7706d9	432MB
docker-sonic-mgmt-framework	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	39ee09777f3e	404MB
docker-router-advertiser	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	2e5c3946b48d	312MB
docker-mux	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	f3d301f161fe	364MB
docker-sonic-gnm	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	a01cd15747bc	403MB
Docker-database	SONiC.D-Link-SONiC_20250731_1.00-3600293-ZTP	36f435b58f7	320MB

[Go Back To Beginning of the document or Beginning of this section](#)





2.1.4.2 Show System Status

This sub-section explains some set of sub-commands that are used to display the status of various parameters pertaining to the physical state of the network node.

show clock

This command displays the current date and time configured on the system

- Usage:

```
show clock
```

- Example:

```
admin@sonic:~$ show clock
Mon Mar 25 20:25:16 UTC 2019
```

config clock date

This command will set the date-time of the systetm, given strings with date-time format HH:MM:SS

- Usage:

```
config clock date <YYYY-MM-DD> <HH:MM:SS>
```

- Parameters:

- **date**: valid date in format YYYY-MM-DD
- **time**: valid time in format HH:MM:SS

- Example:

```
admin@sonic:~$ config clock date 2023-04-10 13:54:36
```

config clock timezone

This command will set the timezone of the systetm, given a string of a valid timezone.

- Usage:

```
config clock timezone <timezone>
```

- Parameters:

- **timezone**: valid timezone to be configured

- Example:

```
admin@sonic:~$ config clock timezone Africa/Accra
```

show clock timezones

This command Will display list of all valid timezones to be configured.

- Usage:

```
show clock timezones
```

- Example:

```
root@host:~$ show clock timezones
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara ...
```

show boot

This command displays the current OS image, the image to be loaded on next reboot, and lists all the available images installed on the device

- Usage:

```
show boot
```

- Example:

```
admin@sonic:~$ show boot Current:
SONiC-OS-20181130.31 Next: SONiC-
OS-20181130.31 Available:
SONiC-OS-20181130.31
```







show environment

This command displays the platform environmental, such as voltages, temperatures and fan speeds

- Usage:

```
show environment
```

- Example:

```
admin@sonic:~$ show environment
coretemp-isa-0000
Adapter: ISA adapter
Package id 0:      +56.0 C    (high = +82.0 C. crit = +104.0 C)
Core 0:           +56.0 C    (high = +82.0 C, crit = +104.0 C)
Core 1:           +56.0 C    (high = +82.0 C, crit = +104.0 C)
Core 2:           +56.0 C    (high = +82.0 C, crit = +104.0 C)
Core 3:           +56.0 C    (high = +82.0 C, crit = +104.0 C)

Dps460-i2c-1-58
Adapter: i2c-0-mux (chan_id 0)
vin:              109.25 V (crit min = +75.00 V. min = +80.00 V)
                  (max = +285.50 V. crit max = +290.50 V)
vount1:           12.08 V (crit min = +9.00 V. min = +290.50 V)|
                  (crit max = +13.75 V)
fan1:             6032 RPM
temp1:            +40.2 C   (low = -35.0 C. high = +60.0 C)
                  (crit low = -40.0 C, crit = +63.0 C)
temp2:            +44.9 C   (low = -35.0 C. high = +60.0 C)
                  (crit low = -40.0 C, crit = +63.0 C)
temp3:            +44.9 C   (low = -35.0 C. high = +60.0 C)
                  (crit low = -40.0 C, crit = +63.0 C)
pin:              67.50 W   (max = 859.00 W)
pount1:           55.06 W   (max = 660.00 W. cap = 550.00W)
...
```

NOTE: The show output has got lot of information; only the sample output is given in the above example. Though the displayed output slightly differs from one platform to another platform, the overall content will be similar to the example mentioned above.

show reboot-cause

This command displays the cause of the previous

- reboot • Usage:

```
show reboot-cause
```

- Example:

2.1.4.3 Shown below is the output of the CLI when executed on the NPU

```
admin@sonic:~$ show reboot-cause
User issued reboot command [User: admin, Time: Mon Mar 25 01:02:03 UTC 2019]
```



2.1.4.4 Shown below is the output of the CLI when executed on the DPU

```
admin@sonic:~$ show reboot-cause reboot
```

show reboot-cause all

This command displays the cause of the previous reboot for the Switch and the DPUs for which the midplane interfaces are up.

- Usage:

```
show reboot-cause all
```

2.1.4.5 Shown below is the output of the CLI when executed on the NPU

- Usage:

```
show reboot-cause all
```

2.1.4.6 Shown below is the output of the CLI when executed on the DPU

```
root@sonic:/home/admin# show reboot-cause all
```

```
Usage: show reboot-cause [OPTIONS] COMMAND [ARGS]...
```

```
Try "show reboot-cause -h" for help.
```

show reboot-cause history

This command displays the history of the previous reboots up to 10 entry

- Usage:

```
show reboot-cause history
```

2.1.4.7 Shown below is the output of the CLI when executed on the NPU

```
root@MtFuji:/home/cisco# show reboot-cause history
```

2.1.4.8 Shown below is the output of the CLI when executed on the DPU

```
root@sonic:/home/admin# show reboot-cause history
```

2.1.4.9 Shown below is the output of the CLI when executed on the NPU

```
# show reboot-cause history all
```

2.1.4.10 Shown below is the output of the CLI when executed on the DPU

```
root@sonic:/home/admin# show reboot-cause history all
```

```
Usage: show reboot-cause history [OPTIONS]
```

```
Try "show reboot-cause history -h" for help.
```

show reboot-cause history DPU1

This command displays the history of the previous reboots up to 10 entry of DPU1. If DPU1 is powered down then there won't be any data in the DB and the "show reboot-cause history DPU1" output will be blank.

- Usage:

```
show reboot-cause history DPU1
```




2.1.4.11 Shown below is the output of the CLI when executed on the NPU

```
# show reboot-cause history DPU1
```

2.1.4.12 Shown below is the output of the CLI when executed on the DPU

```
root@sonic:/home/admin# show reboot-cause history DPU1
```

```
Usage: show reboot-cause history [OPTIONS]
```

```
Try "show reboot-cause history -h" for help.
```

show uptime

This command displays the current system uptime

- Usage:

```
show uptime
```

- Example:

```
admin@sonic:~$ show uptime
```

```
up 2 days, 21 hours, 30 minutes
```

show logging

This command displays all the currently stored log messages. All the latest processes and corresponding transactions are stored in the “syslog” file. This file is saved in the path `/var/log` and can be viewed by giving the command `sudo cat syslog` as this requires root login.

- Usage:

```
show logging [(<process_name> [-l|--lines <number_of_lines>]) | (-f|--follow)]
```

- Example:

```
admin@sonic:~$ show logging
```

It can be useful to pipe the output from `show logging` to the command `more` in order to examine one screenful of log messages at a time

- Example:

```
admin@sonic:~$ show logging | more
```

Optionally, you can specify a process name in order to display only log messages mentioning that process

- Example:

```
admin@sonic:~$ show logging sensord
```

Optionally, you can specify a number of lines to display using the `-l` or `--lines` option. Only the most recent N lines will be displayed. Also note that this option can be combined with a process name.

- Examples:

```
admin@sonic:~$ show logging --lines 50
```

```
admin@sonic:~$ show logging sensord --lines 50
```

Optionally, you can follow the log live as entries are written to it by specifying the `-f` or `--follow` flag

- Example:

```
admin@sonic:~$ show logging --follow
```



show users

This command displays a list of users currently logged in to the device

- Usage:

```
show users
```

- Examples:

```
admin@sonic:~$ show users
admin      pts/9          Mar 25 20:31 (100.127.20.23)

admin@sonic:~$ show users
admin      ttyS1         2019-03-25 20:31
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.4.13 Show Hardware Platform

The information displayed in this set of commands partially overlaps with the one generated by “show environment” instruction. In this case though, the information is presented in a more succinct fashion. In the future these two CLI stanzas may end up getting combined.

show platform summary

This command displays a summary of the device’s hardware platform

- Usage:

```
show platform summary
```

show platform syseeprom

This command displays information stored on the system EEPROM. Note that the output of this command is not the same for all vendor’s platforms. Couple of example outputs are given below.

- Usage:

```
show platform syseeprom
```

show platform ssdhealth

This command displays health parameters of the device’s SSD

- Usage:

```
show platform ssdhealth [--vendor]
```

show platform psustatus

This command displays the status of the device’s power supply units

- Usage:

```
show platform psustatus
```

show platform fan

This command displays the status of the device’s fans

- Usage:

```
show platform fan
```





show platform temperature

This command displays the status of the device's thermal sensors

- Usage:

```
show platform temperature
```

2.1.4.13.1 Transceivers

Displays diagnostic monitoring information of the transceivers

show interfaces transceiver

This command displays information for all the interfaces for the transceiver requested or a specific interface if the optional "interface_name" is specified.

- Usage:

```
show interfaces transceiver (eeprom [-d|--dom] | info | lpmode | presence | error-status [-hw|--fetch-  
from-hardware] | pm | status) [<interface_name>]
```

Example (Decode and display information stored on the EEPROM of SFP transceiver connected to Ethernet0):

```
admin@sonic:~$ show interfaces transceiver eeprom --dom Ethernet0 Ethernet0: SFP  
detected  
Connector : No separable connector Encoding :  
Unspecified  
Extended Identifier : Unknown  
Extended RateSelect Compliance : QSFP+ Rate Select Version 1 Identifier : QSFP+  
Length Cable Assembly(m) : 1  
Specification compliance :  
10/40G Ethernet Compliance Code : 40GBASE-CR4 Fibre  
Channel Speed : 1200 Mbytes/Sec  
Fibre Channel link length/Transmitter Technology : Electrical inter-enclosure (EL)  
Fibre Channel transmission media : Twin Axial Pair (TW) Vendor Date  
Code(YYYY-MM-DD Lot) : 2015-10-31  
Vendor Name : XXXXX Vendor OUI : XX-XX-XX  
Vendor PN : 1111111111  
Vendor Rev :  
Vendor SN : 11111111  
ChannelMonitorValues:  
RX1Power: -1.1936dBm  
RX2Power: -1.1793dBm  
RX3Power: -0.9388dBm  
RX4Power: -1.0729dBm  
TX1Bias: 4.0140mA  
TX2Bias: 4.0140mA  
TX3Bias: 4.0140mA  
TX4Bias: 4.0140mA  
ModuleMonitorValues : Temperature :  
1.1111C Vcc : 0.0000Volts
```



- Example (Display status of low-power mode of SFP transceiver connected to Ethernet~~xx~~):

```
admin@sonic:~$ show interfaces transceiver lpmode Ethernetxx
Port                Low-power Mode
-----
Ethernetxx         On
```

- Example (Display presence of SFP transceiver connected to Ethernet~~xx~~):

```
admin@sonic:~$ show interfaces transceiver presence Ethernetxx
Port                Presence
-----
Ethernetxx         Present
```

- Example (Display error status of SFP transceiver connected to Ethernet~~xx~~):

```
admin@sonic:~$ show interfaces transceiver error-status Ethernetxx
Port                Error Status
-----
Ethernetxx         OK
```

- Example (Display performance monitoring info of SFP transceiver connected to Ethernet~~xx~~):

```
admin@sonic:~$ show interfaces transceiver pm Ethernetxx
```

- Example (Display status info of SFP transceiver connected to Ethernet~~xx~~):

```
admin@sonic:~$ show interfaces transceiver status Ethernetxx
```

[Go Back To Beginning of the document or Beginning of this section](#)





2.1.5 AAA & TACACS+

This section captures the various show commands & configuration commands that are applicable for the AAA (Authentication, Authorization, and Accounting) module. Admins can configure the type of authentication (local or remote tacacs based) required for the users and also the authentication failthrough and fallback options. Following show command displays the current running configuration related to the AAA.

2.1.5.1 AAA

2.1.5.1.1 AAA show commands

This command is used to view the Authentication, Authorization & Accounting settings that are configured in the network node.

show aaa

This command displays the AAA settings currently present in the network node

- Usage:

```
show aaa
```

- Example:

```
admin@sonic:~$ show aaa
AAA authentication login local (default)
AAA authentication failthrough True (default) AAA
authentication fallback True (default)
```

2.1.5.1.2 AAA config commands

This sub-section explains all the possible CLI based configuration options for the AAA module. The list of commands/sub-commands possible for aaa is given below.

Command: aaa authentication sub-commands: - aaa authentication failthrough - aaa authentication fallback - aaa authentication login

aaa authentication failthrough

This command is used to either enable or disable the failthrough option. This command is useful when user has configured more than one tacacs+ server and when user has enabled tacacs+ authentication. When authentication request to the first server fails, this configuration allows to continue the request to the next server. When this configuration is enabled, authentication process continues through all servers configured. When this is disabled and if the authentication request fails on first server, authentication process will stop and the login will be disallowed.

- Usage:

```
config aaa authentication failthrough (enable | disable | default)
```

- Parameters:

- enable: This allows the AAA module to process with local authentication if remote authentication fails.
- disable: This disallows the AAA module to proceed further if remote authentication fails.
- default: This re-configures the default value, which is “enable”.

- Example:

```
admin@sonic:~$ sudo config aaa authentication failthrough enable
```



aaa authentication fallback

The command is not used at the moment. When the tacacs+ authentication fails, it falls back to local authentication by default.

- Usage:

```
config aaa authentication fallback (enable | disable | default)
```

- Example:

```
admin@sonic:~$ sudo config aaa authentication fallback enable
```

aaa authentication login

This command is used to either configure whether AAA should use local database or remote tacacs+ database for user authentication. By default, AAA uses local database for authentication. New users can be added/deleted using the linux commands (Note that the configuration done using linux commands are not preserved during reboot). Admin can enable remote tacacs+ server based authentication by selecting the AUTH_PROTOCOL as tacacs+ in this command. Admins need to configure the tacacs+ server accordingly and ensure that the connectivity to tacacs+ server is available via the management interface. Once if the admins choose the remote authentication based on tacacs+ server, all user logins will be authenticated by the tacacs+ server. If the authentication fails, AAA will check the “failthrough” configuration and authenticates the user based on local database if failthrough is enabled.

- Usage:

```
config aaa authentication (tacacs+ | local | default)
```

- Parameters:

- tacacs+: Enables remote authentication based on tacacs+
- local: Disables remote authentication and uses local authentication
- default: Reset back to default value, which is only “local” authentication

- Example:

```
admin@sonic:~$ sudo config aaa authentication login tacacs+
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)





2.1.5.2 TACACS+

2.1.5.2.1 TACACS+ show commands show tacacs

This command displays the global configuration fields and the list of all tacacs servers and their corresponding configurations.

- Usage:

```
show tacacs
```

- Example:

```
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 99
TACPLUS global passkey <EMPTY_STRING> (default)

TACPLUS_SERVER address 10.11.12.14 priority 9
                        tcp_port 50
                        auth_type mschap
                        timeout 10
                        passkey testing789

TACPLUS_SERVER address 10.0.0.9 priority
                        1 tcp_port
                        49
```

2.1.5.2.2 TACACS+ config commands

This sub-section explains the command “config tacacs” and its sub-commands that are used to configure the following tacacs+ parameters. Some of the parameters like authtype, passkey and timeout can be either configured at per server level or at global level (global value will be applied if there no server level configuration)

1. Add/Delete the tacacs+ server details.
2. authtype - global configuration that is applied to all servers if there is no server specific configuration.
3. default - reset the authtype or passkey or timeout to the default values.
 4. passkey - global configuration that is applied to all servers if there is no server specific configuration.
 5. timeout - global configuration that is applied to all servers if there is no server specific configuration.

config tacacs add



This command is used to add a TACACS+ server to the tacacs server list. Note that more than one tacacs+ (maximum of seven) can be added in the device. When user tries to login, tacacs client shall contact the servers one by one. When any server times out, device will try the next server one by one based on the priority value configured for that server. When this command is executed, the configured tacacs+ server addresses are updated in /etc/pam.d/common-auth-sonic configuration file which is being used by tacacs service.

- Usage:

```
config tacacs add <ip_address> [-t|--timeout <seconds>] [-k|--key <secret>] [-a|--type <type>] [-o|--port <port>] [-p|--pri <priority>] [-m|--use-mgmt-vrf]
```

- Parameters:

- ip_address: TACACS+ server IP address.
- timeout: Transmission timeout interval in seconds, range 1 to 60, default 5
- key: Shared secret
- type: Authentication type, “chap” or “pap” or “mschap” or “login”, default is “pap”.
- port: TCP port range is 1 to 65535, default 49
- pri: Priority, priority range 1 to 64, default 1.
- use-mgmt-vrf: This means that the server is part of Management vrf, default is “no vrf”

- Example:

```
admin@sonic:~$ sudo config tacacs add 10.11.12.13 -t 10 -k testing789 -a mschap -o 50 -p 9
```

- Example Server Configuration in /etc/pam.d/common-auth-sonic configuration file:

```
auth      [success=done new_authtok_reqd=done default=ignore]                pam_tacplus.so
server=10.11.12.14:50 secret=testing789 login=mschap timeout=10 try_first_pass auth[success=done
new_authtok_reqd=done default=ignore]                pam_tacplus.so
server=10.11.12.24:50 secret=testing789 login=mschap timeout=987654321098765433211
0987 try_first_pass
auth      [success=done new_authtok_reqd=done default=ignore]                pam_tacplus.so
server=10.0.0.9:49 secret= login=mschap timeout=5 try_first_pass
auth      [success=done new_authtok_reqd=done default=ignore]                pam_tacplus.so
server=10.0.0.8:49 secret= login=mschap timeout=5 try_first_pass
auth      [success=done new_authtok_reqd=done default=ignore]                pam_tacplus.so
server=10.11.12.13:50 secret=testing789 login=mschap timeout=10 try_first_pass auth[success=1
default=ignore]                pam_unix.so nullok try_first_pass
```

NOTE: In the above example, the servers are stored (sorted) based on the priority value configured for the server.

config tacacs delete

This command is used to delete the tacacs+ servers configured.

- Usage:

```
config tacacs delete <ip_address>
```

- Example:

```
admin@sonic:~$ sudo config tacacs delete 10.11.12.13
```



config tacacs authtype

This command is used to modify the global value for the TACACS+ authtype. When user has not configured server specific authtype, this global value shall be used for that server.

- Usage:

```
config tacacs authtype (chap | pap | mschap | login)
```

- Example:

```
admin@sonic:~$ sudo config tacacs authtype mschap
```

config tacacs default

This command is used to reset the global value for authtype or passkey or timeout to default value. Default for authtype is “pap”, default for passkey is EMPTY_STRING and default for timeout is 5 seconds.

- Usage:

```
config tacacs default (authtype | passkey | timeout)
```

- Example (This will reset the global authtype back to the default value “pap”):

```
admin@sonic:~$ sudo config tacacs default authtype
```

config tacacs passkey

This command is used to modify the global value for the TACACS+ passkey. When user has not configured server specific passkey, this global value shall be used for that server.

- Usage:

```
config tacacs passkey <pass_key>
```

- Example:

```
admin@sonic:~$ sudo config tacacs passkey testing123
```

config tacacs timeout

This command is used to modify the global value for the TACACS+ timeout. When user has not configured server specific timeout, this global value shall be used for that server.

- Usage:

```
config tacacs [default] timeout [<timeout_value_in_seconds>]
```

- Options:

- Valid values for timeout is 1 to 60 seconds.
- When the optional keyword “default” is specified, timeout_value_in_seconds parameter wont be used; default value of 5 is used.
- Configuration using the keyword “default” is introduced in 201904 release.

- Example: To configure non-default timeout value

```
admin@sonic:~$ sudo config tacacs timeout 60
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.6 ACL

This section explains the various show commands and configuration commands available for users.

2.1.6.1 ACL show commands

show acl table

This command displays either all the ACL tables that are configured or only the specified “TABLE_NAME”. Output from the command displays the table name, type of the table, the list of interface(s) to which the table is bound and the description about the table.

- Usage:

```
show acl table [<table_name>]
```

- Example:

```
admin@sonic:~$ show acl table
```

Name	Type	Binding	Description	Stage
EVERFLO W	MIRROR	Ethernet16 Ethernet96 Ethernet108 Ethernet112 PortChannel0001 PortChannel0002	EVERFLO W	ingress
SNMP_AC L	CTRLPLAN E	SNMP	SNMP_AC L	ingress
DT_ACL_T1 L3		Ethernet0 Ethernet4 Ethernet112 Ethernet116	DATA_ACL_TABLE_1	egress
SSH_ONLY	CTRLPLANE	SSH	SSH_ONLY	ingress

show acl rule

This command displays all the ACL rules present in all the ACL tables or only the rules present in specified table “TABLE_NAME” or only the rule matching the RULE_ID option. Output from the command gives the following information about the rules

1. Table name - ACL table name to which the rule belongs to.
2. Rule name - ACL rule name
3. Priority - Priority for this rule.
4. Action - Action to be performed if the packet matches with this ACL rule.

It can be:

- “DROP”/“FORWARD”(“ACCEPT” for control plane ACL)
- “REDIRECT: redirect-object” for redirect rule, where “redirect-object” is either:
 - physical interface name, e.g. “Ethernet10”



- port channel name, e.g. “PortChannel0002”
- next-hop IP address, e.g. “10.0.0.1”
- next-hop group set of IP addresses with comma separator, e.g. “10.0.0.1,10.0.0.3”
- “MIRROR INGRESS|EGRESS: session-name” for mirror rules, where “session-name” refers to mirror session

Users can choose to have a default permit rule or default deny rule. In case of default “deny all” rule, add the permitted rules on top of the deny rule. In case of the default “permit all” rule, users can add the deny rules on top of it. If users have not configured any rule, SONiC allows all traffic (which is “permit all”).

5. Match - The fields from the packet header that need to be matched against the same present in the incoming traffic.

- Usage:

```
show acl rule [<table_name>] [<rule_id>]
```

- Example:

```
admin@sonic:~$ show acl rule
```

Table	Rule	Priority	Action	Match
-----	-----	-----	-----	

SNMP_ACL	RULE_1	9999	ACCEPT	IP_PROTOCOL: 17 SRC_IP:
1.1.1.1/32				
SSH_ONLY	RULE_2	9998	ACCEPT	IP_PROTOCOL: 6 SRC_IP:
1.1.1.1/32				
EVERFLOW	RULE_3	9997	MIRROR INGRESS: everflow0	SRC_IP:
20.0.0.2/32				
EVERFLOW	RULE_4	9996	MIRROR EGRESS : everflow1	L4_SRC_PORT: 4621
DATAACL	RULE_5	9995	REDIRECT: Ethernet8 FORWARD	IP_PROTOCOL: 126
DATAACL	RULE_6	9994	FORWAR	L4_SRC_PORT: 179
DATAACL	RULE_7	9993	D DROP	L4_DST_PORT: 179
SNMP_ACL	DEFAULT_RULE	1	DROP	ETHER_TYPE: 2048
SSH_ONLY	DEFAULT_RULE	1		ETHER_TYPE: 2048

2.1.6.2 ACL config commands

This sub-section explains the list of configuration options available for ACL module. Note that there is no direct command to add or delete or modify the ACL table and ACL rule. Existing ACL tables and ACL rules can be updated by specifying the ACL rules in json file formats and configure those files using this CLI command.

config acl update full

This command is to update the rules in all the tables or in one specific table in full. If a table_name is provided, the operation will be restricted in the specified table. All existing rules in the specified table or all tables will be removed. New rules loaded from file will be installed. If the table_name is specified, only rules within that table will be removed and new rules in that table will be installed. If the table_name is not specified, all rules from all tables will be removed and only the rules present in the input file will be added.



The command does not modify anything in the list of acl tables. It modifies only the rules present in those pre-existing tables.

In order to create acl tables, either follow the config_db.json method or minigraph method to populate the list of ACL tables.

After creating tables, either the config_db.json method or the minigraph method or the CLI method (explained here) can be used to populate the rules in those ACL tables.

This command updates only the ACL rules and it does not disturb the ACL tables; i.e. the output of “show acl table” is not altered by using this command; only the output of “show acl rule” will be changed after this command.

When “-session_name” optional argument is specified, command sets the session_name for the ACL table with this mirror session name. It fails if the specified mirror session name does not exist.

When “-mirror_stage” optional argument is specified, command sets the mirror action to ingress/egress based on this parameter. By default command sets ingress mirror action in case argument is not specified.

When the optional argument “max_priority” is specified, each rule’s priority is calculated by subtracting its “sequence_id” value from the “max_priority”. If this value is not passed, the default “max_priority” 10000 is used.

- Usage:

```
config acl update full [--table_name <table_name>] [--session_name <session_name>] [--mirror_stage (ingress | egress)] [--max_priority <priority_value>] <acl_json_file_name>
```

- Parameters:

- table_name: Specify the name of the ACL table to load. Example: config acl update full “-table_name DT_ACL_T1 /etc/sonic/acl_table_1.json”
- session_name: Specify the name of the ACL session to load. Example: config acl update full “-session_name mirror_ses1 /etc/sonic/acl_table_1.json”
- priority_value: Specify the maximum priority to use when loading ACL rules. Example: config acl update full “-max-priority 100 /etc/sonic/acl_table_1.json”

NOTE 1: All these optional parameters should be inside double quotes. If none of the options are provided, double quotes are not required for specifying filename alone. NOTE 2: Any number of optional parameters can be configured in the same command.

- Examples:

```
admin@sonic:~$ sudo config acl update full /etc/sonic/
acl_full_snmp_1_2_ssh_4.json
admin@sonic:~$ sudo config acl update full "--table_name SNMP-ACL /etc/sonic/
acl_full_snmp_1_2_ssh_4.json"
admin@sonic:~$ sudo config acl update full "--session_name everflow0 /etc/sonic/
acl_full_snmp_1_2_ssh_4.json"
```

This command will remove all rules from all the ACL tables and insert all the rules present in this input file. Refer the example file acl_full_snmp_1_2_ssh_4.json that adds two rules for SNMP (Rule1 and Rule2) and one rule for SSH (Rule4) Refer an example for input file format here Refer another example here



config acl update incremental

This command is used to perform incremental update of ACL rule table. This command gets existing rules from Config DB and compares with rules specified in input file and performs corresponding modifications.

With respect to DATA ACLs, the command does not assume that new dataplane ACLs can be inserted in between by shifting existing ACLs in all ASICs. Therefore, this command performs a full update on dataplane ACLs. With respect to control plane ACLs, this command performs an incremental update. If we assume that “file1.json” is the already loaded ACL rules file and if “file2.json” is the input file that is passed as parameter for this command, the following requirements are valid for the input file.

1. First copy the file1.json to file2.json.
2. Remove the unwanted ACL rules from file2.json
3. Add the newly required ACL rules into file2.json.
4. Modify the existing ACL rules (that require changes) in file2.json.

NOTE: If any ACL rule that is already available in file1.json is required even after this command execution, such rules should remain unaltered in file2.json. Don't remove them. Note that “incremental” is working like “full”.

When “-session_name” optional argument is specified, command sets the session_name for the ACL table with this mirror session name. It fails if the specified mirror session name does not exist.

When “-mirror_stage” optional argument is specified, command sets the mirror action to ingress/egress based on this parameter. By default command sets ingress mirror action in case argument is not specified.

When the optional argument “max_priority” is specified, each rule's priority is calculated by subtracting its “sequence_id” value from the “max_priority”. If this value is not passed, the default “max_priority” 10000 is used.

- Usage:

```
config acl update incremental [--session_name <session_name>] [--mirror_stage (ingress | egress)] [--max_priority <priority_value>] <acl_json_file_name>
```

- Parameters:

- table_name: Specify the name of the ACL table to load. Example: config acl update full “-table_name DT_ACL_T1 /etc/sonic/acl_table_1.json”
- session_name: Specify the name of the ACL session to load. Example: config acl update full “-session_name mirror_ses1 /etc/sonic/acl_table_1.json”
- priority_value: Specify the maximum priority to use when loading ACL rules. Example: config acl update full “-max-priority 100 /etc/sonic/acl_table_1.json”

NOTE 1: All these optional parameters should be inside double quotes. If none of the options are provided, double quotes are not required for specifying filename alone. NOTE 2: Any number of optional parameters can be configured in the same command.

- Examples:

```
admin@sonic:~$ sudo config acl update incremental /etc/sonic/acl_incremental_snmp_1_3_ssh_4.json
```




```
admin@sonic:~$ sudo config acl update incremental "--session_name everflow0 /
etc/sonic/acl_incremental_snmp_1_3_ssh_4.json"
```

Refer the example file `acl_incremental_snmp_1_3_ssh_4.json` that adds two rules for SNMP (Rule1 and Rule3) and one rule for SSH (Rule4). When this “incremental” command is executed after “full” command, it has removed SNMP Rule2 and added SNMP Rule3 in the example. File “`acl_full_snmp_1_2_ssh_4.json`” has got SNMP Rule1, SNMP Rule2 and SSH Rule4. File “`acl_incremental_snmp_1_3_ssh_4.json`” has got SNMP Rule1, SNMP Rule3 and SSH Rule4. This file is created by copying the file “`acl_full_snmp_1_2_ssh_4.json`” to “`acl_incremental_snmp_1_3_ssh_4.json`” and then removing SNMP Rule2 and adding SNMP Rule3.

config acl add table

This command is used to create new ACL

tables. • Usage:

```
config acl add table [OPTIONS] <table_name> <table_type> [-d <description>] [-p <ports>] [-s (ingress |
egress)]
```

• Parameters:

- `table_name`: The name of the ACL table to create.
- `table_type`: The type of ACL table to create (e.g. “L3”, “L3V6”, “L3V4V6”, “MIRROR”)
- `description`: A description of the table for the user. (default is the `table_name`)
- `ports`: A comma-separated list of ports/interfaces to add to the table. The behavior is as follows:
 - Physical ports will be bound as physical ports
 - Portchannels will be bound as portchannels - passing a portchannel member is invalid
 - VLANs will be expanded into their members (e.g. “Vlan1000” will become “Ethernet0,Ethernet2,Ethernet4...”)
- `stage`: The stage this ACL table will be applied to, either ingress or egress. (default is ingress)

ingress) • Examples:

```
admin@sonic:~$ sudo config acl add table EXAMPLE L3 -p Ethernet0,Ethernet4 -s ingress
```

```
admin@sonic:~$ sudo config acl add table EXAMPLE_2 L3V6 -p
Vlan1000,PortChannel0001,Ethernet128 -s egress
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

aclshow

This command is used to display: ACL rules, tables and their priority, ACL packets counters, and bytes counters

• Usage:

```
aclshow [-h] [-a] [-c] [-r RULES] [-t TABLES] [-v] [-vv]
```

• Parameters:

- `-a, -all`: Show all ACL counters
- `-c, -clear`: Clear ACL counters statistics



- -r RULES, -rules RULES: Show only specified ACL rules and their counters
- -t TABLES, -tables TABLES: Show only specified ACL tables and their counters
- -vv, -verbose: Verbose output
- Examples:

```
admin@sonic:~$ sudo aclshow -a
```

RULE NAME	TABLE NAME	PRIO	PACKETS COUNT	BYTES COUNT
-----	-----	-----	-----0	-----0
RULE_1 RULE_2	DATAACL	9999	0	0
RULE_1	DATAACL	9998	N/A	N/A
	SNMP_ACL	9999		

If the `PACKETS COUNT` and `BYTES COUNT` fields have the `N/A` value it means either that the ACL rule is invalid or it is a control plane ACL and those counters are created in Linux, not in SONiC `COUNTERS_DB` and the `iptables` utility should be used to view those counters.

If the `PACKETS COUNT` and `BYTES COUNT` fields have some numeric value it means that it is a SONiC ACL's and those counters are created in SONiC `COUNTERS_DB`.

2.1.7 ASIC SDK health event

2.1.7.1 ASIC SDK health event config commands

****config asic-sdk-health-event suppress ****

This command is for a customer to configure the categories that he/she wants to suppress for a certain severity.

- Usage:

```
config config asic-sdk-health-event suppress <severity> [--category-list <category-list>|<none>|<all>] [--max-events <max-events>]
```

- Parameters:

- severity: Specify the severity whose ASIC/SDK health events to be suppressed. It can be one of `fatal`, `warning`, and `notice`.
- category-list: Specify the categories from which the ASIC/SDK health events to be suppressed. It is a list whose element is one of `software`, `firmware`, `cpu_hw`, `asic_hw` separated by a comma. If the category-list is `none`, none category is suppressed and all the categories will be notified for `severity`. In this case, it will not be stored in the `CONFIG_DB`. If the category-list is `all`, all the categories are suppressed and none category will be notified for `severity`.
- max-events: Specify the maximum number of events of the severity to be stored in the `STATE_DB`. There is no limitation if the max-events is 0. In this case, it will not be stored in the `CONFIG_DB`.

- Examples:

```
admin@sonic:~$ sudo config asic-sdk-health-event suppress fatal --category-list cpu_hw,software --max-events 10240
```

This command will suppress ASIC/SDK health events whose severity is `fatal` and category is `cpu_hw` or `software`. Maximum number of such events in the `STATE_DB` is 10240.

2.1.7.2 ASIC SDK health event show commands

show asic-sdk-health-event received



This command displays the received ASIC/SDK health

events. • Usage:

```
show asic-sdk-health-event received [-n <asicname>]
```

• Details:

- show asic-sdk-health-event received: Display the ASIC/SDK health events received on all ASICs
- show asic-sdk-health-event received -n asic0: Display all the ASIC/SDK health events received on asic0

asic0 •

Example:

```
admin@sonic:~$ show asic-sdk-health-event received
Time                Severity      Category --- Description
-----2023-        -----fatal -----
10-20 05:07:34 2023-10-    fatal      firmware   Command timeout
20 03:06:25 2023-10-20    fatal      software   SDK daemon keep alive failed
05:07:34                notice     asic_hw    Uncorrectable ECC error
2023-10-20 01:58:43                asic_hw    Correctable ECC error
```

• Example on a multi ASIC system:

```
admin@sonic:~$ show asic-sdk-health-event received
asic0:
Time                Severity      Category --- Description
-----2023-        -----fatal -----
10-20 05:07:34 2023-10-    fatal      firmware   Command timeout
20 03:06:25 asic1:                software   SDK daemon keep alive failed
Time                Severity
-----2023-        -----fatal Category --- Description
10-20 05:07:34    notice     -----asic_hw -----Uncorrectable
2023-10-20 01:58:43                asic_hw    ECC error
Correctable ECC error
```

Optionally, you can specify the asic name in order to display the ASIC/SDK health events received on that particular ASIC on a multi ASIC system

• Example:

```
admin@sonic:~$ show asic-sdk-health-event received -n asic1
asic1:
Time                Severity      Category --- Description
-----
2023-10-20 05:07:34    fatal      firmware   Command timeout
```

show asic-sdk-health-event suppress-configuration

This command displays the suppressed category list and maximum number of events of ASIC/SDK health events.

• Usage:

```
show asic-sdk-health-event suppressed-category-list [-n <asicname>]
```

• Details:



- show asic-sdk-health-event suppress-configuration: Display the ASIC/SDK health event suppress category list and maximum number of events on all ASICs
- show asic-sdk-health-event suppress-configuration -n asic0: Display all the ASIC/SDK health event suppress category list and maximum number of events on asic0
- Example:

```
admin@sonic:~$ show asic-sdk-health-event suppress-configuration
Severity      Suppressed category-list      Max events
-----fatal  -----software -----
notice        none                          unlimited 1024
warning       firmware,asic_hw              10240
```

- Example on a multi ASIC system:

```
admin@sonic:~$ show asic-sdk-health-event suppress-configuration
asic0:
Severity      Suppressed category-list      Max events
-----
notice        firmware,asic_hw              10240
warning
warning asic1:
Severity      Suppressed category-list      Max events
-----
fatal         software                      unlimited
```

Optionally, you can specify the asic name in order to display the ASIC/SDK health event suppress category list on that particular ASIC on a multi ASIC system

- Example:

```
admin@sonic:~$ show asic-sdk-health-event suppress-configuration -n asic1
asic1:
Severity      Suppressed category-list      Max events
-----
fatal         software                      unlimited
```

2.1.7.3 ASIC SDK health event clear commands

sonic-clear asic-sdk-health-event

This command clears all the received ASIC/SDK health events.

- Usage:

```
sonic-clear asic-sdk-health-event [-n <asicname>]
```

- Details:

- sonic-clear asic-sdk-health-event: Clear the ASIC/SDK health events received on all ASICs
- sonic-clear asic-sdk-health-event -n asic0: Display all the ASIC/SDK health events received on asic0

- Example:

```
admin@sonic:~$ sonic-clear asic-sdk-health-event
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.8 ARP & NDP

2.1.8.1 ARP show commands

show arp

This command displays the ARP entries in the device with following options.

1. Display the entire table.
2. Display the ARP entries learnt on a specific interface.
3. Display the ARP of a specific ip-address.

- Usage:

```
show arp [-if <interface_name>] [<ip_address>]
```

- Details:

- show arp: Displays all entries
- show arp -if : Displays the ARP specific to the specified interface.
- show arp : Displays the ARP specific to the specified ip-address.

- Example:

```
admin@sonic:~$ show arp
```

Address	MacAddress	Iface	Vlan
-----	-----	-----	-----
192.168.1.183	88:5a:92:fb:bf:41	Ethernet44	-
192.168.1.175	88:5a:92:fc:95:81	Ethernet28	-
192.168.1.181	e4:c7:22:c1:07:7c	Ethernet40	-
192.168.1.179	88:5a:92:de:a8:bc	Ethernet36	-
192.168.1.118	00:1c:73:3c:de:43	Ethernet24	-
192.168.1.11	00:1c:73:3c:e1:38	Ethernet38	-
192.168.1.161	24:e9:b3:71:3a:01	Ethernet0	-
192.168.1.189	24:e9:b3:9d:57:41	Ethernet56	-
192.168.1.187	74:26:ac:8b:8f:c1	Ethernet52	-
192.168.1.165	88:5a:92:de:a0:7c	Ethernet8	-

Total number of entries 10

Optionally, you can specify the interface in order to display the ARPs learnt on that particular interface

- Example:

```
admin@sonic:~$ show arp -if Ethernet40
```

Address	MacAddress	Iface	Vlan
-----	-----	-----	-----
192.168.1.181	e4:c7:22:c1:07:7c	Ethernet40	-

Total number of entries 1

Optionally, you can specify an IP address in order to display only that particular entry

- Example:

```
admin@sonic:~$ show arp 192.168.1.181
```

Address	MacAddress	Iface	Vlan
-----	-----	-----	-----
192.168.1.181	e4:c7:22:c1:07:7c	Ethernet40	-

Total number of entries 1



2.1.8.2 NDP show commands

show ndp

This command displays either all the IPv6 neighbor mac addresses, or for a particular IPv6 neighbor, or for all IPv6 neighbors reachable via a specific interface.

- Usage:

```
show ndp [-if|--iface <interface_name>] <ipv6_address>
```

- Example (show all IPv6 neighbors):

```
admin@sonic:~$ show ndp
```

Address	MacAddress	Iface	Vlan	Status
-----	-----	-----	-----	-----
fe80::20c:29ff:feb8:b11e	00:0c:29:b8:b1:1e	eth0	-	REACHABLE
fe80::20c:29ff:feb8:cff0	00:0c:29:b8:cf:f0	eth0	-	REACHABLE
fe80::20c:29ff:feb9:324	00:0c:29:f9:03:24	eth0	-	REACHABLE

Total number of entries 3

- Example (show specific IPv6 neighbor):

```
admin@sonic:~$ show ndp fe80::20c:29ff:feb8:b11e
```

Address	MacAddress	Iface	Vlan	Status
-----	-----	-----	-----	-----
fe80::20c:29ff:feb8:b11e	00:0c:29:b8:b1:1e	eth0	-	REACHABLE

Total number of entries 1

- Example (show IPv6 neighbors learned on a specific interface):

```
admin@sonic:~$ show ndp -if eth0
```

Address	MacAddress	Iface	Vlan	Status
-----	-----	-----	-----	-----
fe80::20c:29ff:feb8:b11e	00:0c:29:b8:b1:1e	eth0	-	REACHABLE
fe80::20c:29ff:feb8:cff0	00:0c:29:b8:cf:f0	eth0	-	REACHABLE
fe80::20c:29ff:feb9:324	00:0c:29:f9:03:24	eth0	-	REACHABLE

Total number of entries 3

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



```
>> show bfd summary
```

```
Total number of BFD sessions: 3
```

Peer Addr	Interface	Vrf	State	Type	Local Addr	TX
Interval	RX Interval	Multiplier	Multihop			
10.0.1.1	default	default	DOWN	async_active	10.0.0.1 300	
500	3 true					
10.0.2.1	Ethernet12	default	UP	async_active	10.0.0.1 200	
600	3 false					
2000::10:1	default	default	UP	async_active	2000::1 100	
	700	3 false				

show bfd peer

This command displays the state and key parameters of all BFD sessions that match an IP address.

- Usage:

```
show bfd peer <peer-ip>
```

- Example:

```
>> show bfd peer 10.0.1.1
```

```
Total number of BFD sessions for peer IP 10.0.1.1: 1
```

Peer Addr	Interface	Vrf	State	Type	Local Addr	TX
Interval	RX Interval	Multiplier	Multihop			
10.0.1.1	default	default	DOWN	async_active	10.0.0.1 300	
500	3 true					

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.10 BGP

This section explains all the BGP show commands and BGP configuration commands in both “Quagga” and “FRR” routing software that are supported in SONiC. In 201811 and older versions “Quagga” was enabled by default. In current version “FRR” is enabled by default. Most of the FRR show commands start with “show bgp”. Similar commands in Quagga starts with “show ip bgp”. All sub-options supported in all these show commands are common for FRR and Quagga. Detailed show commands examples for Quagga are provided at the end of this document. This section captures only the commands supported by FRR.

2.1.10.1 BGP show commands

show bgp summary (Versions >= 201904 using default FRR routing stack)

show ip bgp summary (Versions <= 201811 using Quagga routing stack)

This command displays the summary of all IPv4 & IPv6 bgp neighbors that are configured and the corresponding states.

- Usage:

Versions >= 201904 using default FRR routing stack

```
show bgp summary
```

Versions <= 201811 using Quagga routing stack

```
show ip bgp summary
```

- Example:

```
admin@sonic:~$ show ip bgp summary
```

IPv4 Unicast Summary:

BGP router identifier 10.1.0.32, local AS number 65100 vrf-id 0 BGP table version 6465

RIB entries 12807, using 2001 KiB of memory Peers 4, using
83 KiB of memory

Peer groups 2, using 128 bytes of memory

Neighbor NeighborName	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/ PfxRcd
10.0.0.57 6400 Lab-T1-01	4	64600	3995	4001		0	0	0 00:39:32	
10.0.0.59 6400 Lab-T1-02	4	64600	3995	3998		0	0	0 00:39:32	
10.0.0.61 6400 Lab-T1-03	4	64600	3995	4001		0	0	0 00:39:32	
10.0.0.63	4	64600	3995	3998		0	0	0 00:39:32	

6400 NotAvailable

Total number of neighbors 4



- Example:

```
admin@sonic:~$ show bgp summary
```

IPv4 Unicast Summary:

BGP router identifier 10.1.0.32, local AS number 65100 vrf-id 0 BGP table version 6465

RIB entries 12807, using 2001 KiB of memory Peers 4, using

83 KiB of memory

Peer groups 2, using 128 bytes of memory

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/ PfxRcd
10.0.0.57	4	64600	3995	4001		0	0	0 00:39:32	
6400									
10.0.0.59	4	64600	3995	3998		0	0	0 00:39:32	
6400									
10.0.0.61	4	64600	3995	4001		0	0	0 00:39:32	
6400									
10.0.0.63	4	64600	3995	3998		0	0	0 00:39:32	
6400									

Total number of neighbors 4

IPv6 Unicast Summary:

BGP router identifier 10.1.0.32, local AS number 65100 vrf-id 0 BGP table version

12803

RIB entries 12805, using 2001 KiB of memory Peers 4, using

83 KiB of memory

Peer groups 2, using 128 bytes of memory

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/ PfxRcd
fc00::72	4	64600	3995	5208		0	0	0 00:39:30	
6400									
fc00::76	4	64600	3994	5208		0	0	0 00:39:30	
6400									
fc00::7a	4	64600	3993	5208		0	0	0 00:39:30	
6400									
fc00::7e	4	64600	3993	5208		0	0	0 00:39:30	
6400									

Total number of neighbors 4



Click here to see the example for “show ip bgp summary” for Quagga.

show bgp neighbors (Versions >= 201904 using default FRR routing stack)

show ip bgp neighbors (Versions <= 201811 using Quagga routing stack)

This command displays all the details of IPv4 & IPv6 BGP neighbors when no optional argument is specified.

When the optional argument IPv4_address is specified, it displays the detailed neighbor information about that specific IPv4 neighbor.

Command has got additional optional arguments to display only the advertised routes, or the received routes, or all routes.

In order to get details for an IPv6 neighbor, use “show bgp ipv6 neighbor <ipv6_address>”

command. • Usage:

Versions >= 201904 using default FRR routing stack

```
show bgp neighbors [<ipv4-address> [advertised-routes | received-routes | routes]]
```

Versions <= 201811 using Quagga routing stack

```
show ip bgp neighbors [<ipv4-address> [advertised-routes | received-routes | routes]]
```

• Example:

```
admin@sonic:~$ show bgp neighbors
BGP neighbor is 10.0.0.57, remote AS 64600, local AS 65100, external link Description: DQS5K01T1
BGP version 4, remote router ID 100.1.0.29, local router ID 10.1.0.32 BGP state = Established,
up for 00:42:15
```



```

Last read 00:00:00, Last write 00:00:03
Hold time is 10, keepalive interval is 3 seconds
Configured hold time is 10, keepalive interval is 3 seconds Neighbor capabilities:
  4 Byte AS: advertised and received AddPath:
    IPv4 Unicast: RX advertised IPv4 Unicast and received Route refresh:
      advertised and received(new)
    Address Family IPv4 Unicast: advertised and received
    Hostname Capability: advertised (name: sonic-z9264f-9251, domain name: n/a) not received
    Graceful Restart Capability: advertised and received Remote Restart
      timer is 300 seconds
    Address families by peer: none
Graceful restart information: End-of-RIB
  send: IPv4 Unicast
  End-of-RIB received: IPv4 Unicast Message
statistics:
  Inq depth is 0 Outq
  depth is 0

                                Sent      Rcvd
Opens:                          2        1
Notifications:                  2      0 Updates:
3206                            3202 Keepalives:
845    847 Route Refresh:       0        0
Capability:                      0      0 Total:
4055                            4050

Minimum time between advertisement runs is 0 seconds

For address family: IPv4 Unicast Update
group 1, subgroup 1 Packet Queue length 0
Inbound soft reconfiguration allowed
Community attribute sent to this neighbor(all) 6400 accepted
prefixes

Connections established 1; dropped 0
Last reset 00:42:37, due to NOTIFICATION sent (Cease/Connection collision resolution)
Local host: 10.0.0.56, Local port: 179 Foreign host: 10.0.0.57,
Foreign port: 46419 Nexthop: 10.0.0.56
Nexthop global: fc00::71
Nexthop local: fe80::2204:fff:fe36:9449 BGP
connection: shared network
BGP Connect Retry Timer in Seconds: 120 Read
thread: on  Write thread: on

```

Optionally, you can specify an IP address in order to display only that particular neighbor. In this mode, you can optionally specify whether you want to display all routes advertised to the specified neighbor, all routes received from the specified neighbor or all routes (received and accepted) from the specified neighbor.



- Example:

```
admin@sonic:~$ show bgp neighbors 10.0.0.57
admin@sonic:~$ show bgp neighbors 10.0.0.57 advertised-routes
admin@sonic:~$ show bgp neighbors 10.0.0.57 received-routes
admin@sonic:~$ show bgp neighbors 10.0.0.57 routes
```

Click here to see the example for “show ip bgp neighbors” for Quagga.

****show ip bgp network [[] [(bestpath | multipath | longer-prefixes | json)]]**

This command displays all the details of IPv4 Border Gateway Protocol (BGP) prefixes.

- Usage:

```
show ip bgp network [[<ipv4-address>|<ipv4-prefix>] [(bestpath | multipath | longer-prefixes | json)]]
```

- Example:

NOTE: The “longer-prefixes” option is only available when a network prefix with a “/” notation is used.

```
admin@sonic:~$ show ip bgp network
admin@sonic:~$ show ip bgp network 10.1.0.32 bestpath
admin@sonic:~$ show ip bgp network 10.1.0.32 multipath
admin@sonic:~$ show ip bgp network 10.1.0.32 json
admin@sonic:~$ show ip bgp network 10.1.0.32/32 bestpath
admin@sonic:~$ show ip bgp network 10.1.0.32/32 multipath
admin@sonic:~$ show ip bgp network 10.1.0.32/32 json
admin@sonic:~$ show ip bgp network 10.1.0.32/32 longer-prefixes
```

show bgp ipv6 summary (Versions >= 201904 using default FRR routing stack)

show ipv6 bgp summary (Versions <= 201811 using Quagga routing stack)

This command displays the summary of all IPv6 bgp neighbors that are configured and the corresponding states.

- Usage:

Versions >= 201904 using default FRR routing stack

```
show bgp ipv6 summary
```

Versions <= 201811 using Quagga routing stack



```
show ipv6 bgp summary
```

- Example:

```
admin@sonic:~$ show bgp ipv6 summary
BGP router identifier 10.1.0.32, local AS number 65100 vrf-id 0 BGP table version
12803
RIB entries 12805, using 2001 KiB of memory Peers 4, using
83 KiB of memory
Peer groups 2, using 128 bytes of memory
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/ PfxRcd
NeighborName									
fc00::72	4	64600	3995	5208		0	0	0 00:39:30	
6400 Lab-T1-01									
fc00::76	4	64600	3994	5208		0	0	0 00:39:30	
6400 Lab-T1-02									
fc00::7a	4	64600	3993	5208		0	0	0 00:39:30	
6400 Lab-T1-03									
fc00::7e	4	64600	3993	5208		0	0	0 00:39:30	
6400 Lab-T1-04									

Total number of neighbors 4

[Click here to see the example for “show ipv6 bgp summary” for Quagga.](#)

show bgp ipv6 neighbors (Versions >= 201904 using default FRR routing stack)

show ipv6 bgp neighbors (Versions <= 201811 using Quagga routing stack)

This command displays all the details of one particular IPv6 Border Gateway Protocol (BGP) neighbor. Option is also available to display only the advertised routes, or the received routes, or all routes.

- Usage:

Versions >= 201904 using default FRR routing stack

```
show bgp ipv6 neighbors [<ipv6-address> [(advertised-routes | received-routes | routes)]]
```

Versions <= 201811 using Quagga routing stack

```
show ipv6 bgp neighbors [<ipv6-address> [(advertised-routes | received-routes | routes)]]
```

- Example:

```
admin@sonic:~$ show bgp ipv6 neighbors fc00::72 advertised-routes

admin@sonic:~$ show bgp ipv6 neighbors fc00::72 received-routes

admin@sonic:~$ show bgp ipv6 neighbors fc00::72 routes
```

[Click here to see the example for “show ip bgp summary” for Quagga.](#)

****show ipv6 bgp network [[]] [(bestpath | multipath | longer-prefixes | json)]]**



This command displays all the details of IPv6 Border Gateway Protocol (BGP)

prefixes. • Usage:

```
show ipv6 bgp network [[<ipv6-address>|<ipv6-prefix>] [(bestpath | multipath | longer-prefixes | json)]]
```

- Example:

NOTE: The “longer-prefixes” option is only available when a network prefix with a “/” notation is used.

```
admin@sonic:~$ show ipv6 bgp network

admin@sonic:~$ show ipv6 bgp network fc00::72 bestpath

admin@sonic:~$ show ipv6 bgp network fc00::72 multipath

admin@sonic:~$ show ipv6 bgp network fc00::72 json

admin@sonic:~$ show ipv6 bgp network fc00::72/64 bestpath

admin@sonic:~$ show ipv6 bgp network fc00::72/64 multipath

admin@sonic:~$ show ipv6 bgp network fc00::72/64 json

admin@sonic:~$ show ipv6 bgp network fc00::72/64 longer-prefixes
```

show route-map

This command displays the routing policy that takes precedence over the other route processes that are configured.

- Usage:

```
show route-map
```

- Example:

```
admin@sonic:~$ show route-map
ZEBRA:
route-map RM_SET_SRC, permit, sequence 10 Match
  clauses:
  Set clauses:
    src 10.12.0.102 Call
  clause: Action:
    Exit routemap
ZEBRA:
route-map RM_SET_SRC6, permit, sequence 10 Match
  clauses:
  Set clauses:
    src fc00:1::102 Call
  clause: Action:
    Exit routemap
```



```

BGP:
route-map FROM_BGP_SPEAKER_V4, permit, sequence 10 Match
  clauses:
  Set clauses: Call
  clause: Action:
    Exit routemap
BGP:
route-map TO_BGP_SPEAKER_V4, deny, sequence 10 Match
  clauses:
  Set clauses: Call
  clause: Action:
    Exit routemap
BGP:
route-map ISOLATE, permit, sequence 10 Match
  clauses:
  Set clauses:
    as-path prepend 65000 Call
  clause:
  Action:
    Exit routemap

```

show suppress-fib-pending

This command is used to show the status of suppress pending FIB feature. When enabled, BGP will not advertise routes which aren't yet ofloaded.

- Usage:

```
show suppress-fib-pending
```

- Examples:

```

admin@sonic:~$ show suppress-fib-pending Enabled
admin@sonic:~$ show suppress-fib-pending Disabled

```

show bgp device-global

This command displays BGP device global configuration.

- Usage:

```
show bgp device-global
```

- Options:

- *-j, -json*: display in JSON format

- Example:

```
admin@sonic:~$ show bgp device-global TSA
```

Go Back To Beginning of the document or Beginning of this section





2.1.10.2 BGP config commands

This sub-section explains the list of configuration options available for BGP module for both IPv4 and IPv6 BGP neighbors.

config bgp shutdown all

This command is used to shutdown all the BGP IPv4 & IPv6 sessions. When the session is shutdown using this command, BGP state in “show ip bgp summary” is displayed as “Idle (Admin)”

- Usage:

```
config bgp shutdown all
```

- Example:

```
admin@sonic:~$ sudo config bgp shutdown all
```

config bgp shutdown neighbor

This command is to shut down a BGP session with a neighbor by that neighbor’s IP address or hostname

- Usage:

```
sudo config bgp shutdown neighbor (<ip_address> | <hostname>)
```

- Examples:

```
admin@sonic:~$ sudo config bgp shutdown neighbor 192.168.1.124
```

```
admin@sonic:~$ sudo config bgp shutdown neighbor SONIC02SPINE
```

config bgp startup all

This command is used to start up all the IPv4 & IPv6 BGP

neighbors • Usage:

```
config bgp startup all
```

- Example:

```
admin@sonic:~$ sudo config bgp startup all
```

config bgp startup neighbor

This command is used to start up the particular IPv4 or IPv6 BGP neighbor using either the IP address or hostname.

- Usage:



```
config bgp startup neighbor (<ip-address> | <hostname>)
```

- Examples:

```
admin@sonic:~$ sudo config bgp startup neighbor 192.168.1.124
```

```
admin@sonic:~$ sudo config bgp startup neighbor SONIC02SPINE
```

config bgp remove neighbor

This command is used to remove particular IPv4 or IPv6 BGP neighbor configuration using either the IP address or hostname.

- Usage:

```
config bgp remove neighbor <neighbor_ip_or_hostname>
```

- Examples:

```
admin@sonic:~$ sudo config bgp remove neighbor 192.168.1.124
```

```
admin@sonic:~$ sudo config bgp remove neighbor 2603:10b0:b0f:346::4a
```

```
admin@sonic:~$ sudo config bgp remove neighbor SONIC02SPINE
```

config suppress-fib-pending

This command is used to enable or disable announcements of routes not yet installed in the HW. Once enabled, BGP will not advertise routes which aren't yet ofloaded.

- Usage:

```
config suppress-fib-pending <enabled|disabled>
```

- Examples:

```
admin@sonic:~$ sudo config suppress-fib-pending enabled
```

```
admin@sonic:~$ sudo config suppress-fib-pending disabled
```

config bgp device-global tsa/w-ecmp

This command is used to manage BGP device global configuration.

Feature list:

1. TSA - Traffic-Shift-Away
2. W-ECMP - Weighted-Cost Multi-Path

- Usage:

```
config bgp device-global tsa <enabled|disabled> config bgp device-global w-ecmp <enabled|disabled>
```



- Examples:

```
admin@sonic:~$ config bgp device-global tsa enabled admin@sonic:~$
config bgp device-global w-ecmp enabled
```

Go Back To Beginning of the document or Beginning of this section

2.1.11 Console

This section explains all Console show commands and configuration options that are supported in SONiC.

All commands are used only when SONiC is used as console switch.

All commands under this section are not applicable when SONiC used as regular switch.

2.1.11.1 Console show commands

show line

This command displays serial port or a virtual network connection

status. • Usage:

```
show line (-b|--breif)
```

- Example:

```
admin@sonic:~$ show line
```

Line	Baud -	Flow Control	PID -	Start Time ----	Device --
1	9600	Enabled	-	-	switch1
2	-3	Disabled	-	-	
4	-	Disabled	-	-	
5	-	Disabled	-	-	
		Disabled	-	-	

Optionally, you can display configured console ports only by specifying the `-b` or `--breif` flag. •

Example:

```
admin@sonic:~$ show line -b
```

Line	Baud -	Flow Control	PID -	Start Time ----	Device --
1	9600	Enabled	-	-	switch1



2.1.11.2 Console config commands

This sub-section explains the list of configuration options available for console management module.

config console enable

This command is used to enable SONiC console switch feature.

- Usage:

```
config console enable
```

- Example:

```
admin@sonic:~$ sudo config console enable
```

config console disable

This command is used to disable SONiC console switch feature.

- Usage:

```
config console disable
```

- Example:

```
admin@sonic:~$ sudo config console disable
```

config console add

This command is used to add a console port setting.

- Usage:

```
config console add <port_name> [--baud|-b <baud_rate>] [--flowcontrol|-f] [--devicename|-d  
<remote_device>]
```

- Example:

```
admin@sonic:~$ config console add 1 --baud 9600 --devicename switch1
```



config console del

This command is used to remove a console port setting.

- Usage:

```
config console del <port_name>
```

- Example:

```
admin@sonic:~$ sudo config console del 1
```

config console remote_device

This command is used to update the remote device name for a console port.

- Usage:

```
config console remote_device <port_name> <remote_device>
```

- Example:

```
admin@sonic:~$ sudo config console remote_device 1 switch1
```

config console baud

This command is used to update the baud rate for a console port.

- Usage:

```
config console baud <port_name> <baud_rate>
```

- Example:

```
admin@sonic:~$ sudo config console baud 1 9600
```



config console flow_control

This command is used to enable or disable flow control feature for a console port.

- Usage:

```
config console flow_control {enable|disable} <port_name>
```

- Example:

```
admin@sonic:~$ sudo config console flow_control enable 1
```

2.1.11.3 Console connect commands

connect line

This command allows user to connect to a remote device via console line with an interactive cli.

- Usage:

```
connect line <target> (-d|--devicename)
```

By default, the target is `port_name`.

- Example:

```
admin@sonic:~$ connect line 1 Successful
connection to line 1 Press ^A ^X to
disconnect
```

Optionally, you can connect with a remote device name by specifying the `-d` or `--devicename` flag.

- Example:

```
admin@sonic:~$ connect line --devicename switch1 Successful
connection to line 1
Press ^A ^X to disconnect
```

connect device

This command allows user to connect to a remote device via console line with an interactive cli.

- Usage:

```
connect device <devicename>
```

The command is same with `connect line --devicename <devicename>`

- Example:

```
admin@sonic:~$ connect line 1 Successful
connection to line 1 Press ^A ^X to
disconnect
```



2.1.11.4 Console clear commands

sonic-clear line

This command allows user to connect to a remote device via console line with an interactive cli.

- Usage:

```
sonic-clear line <target> (-d|--devicename)
```

By default, the target is `port_name`.

- Example:

```
admin@sonic:~$ sonic-clear line 1
```

Optionally, you can clear with a remote device name by specifying the `-d` or `--devicename` flag.

- Example:

```
admin@sonic:~$ sonic-clear --devicename switch1
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.11.5 DPU serial console utility

dpu-tty.py

This command allows user to connect to a DPU serial console via TTY device with interactive CLI program: `picocom`. The configuration is from `platform.json`. The utility works only on smart switch that provides DPU UART connections through `/dev/ttyS*` devices.

- Usage:

```
dpu-tty.py (-n|--name) <DPU_NAME> [(-b|--baud) <BAUD_RATE>] [(-t|--tty) <TTY>]
```

- Example:

```
# dpu-tty.py -n dpu0 picocom v3.1

port is flowcontrol : /dev/ttyS4 :
baudrate is parity  none
is databits are    : 115200 :
stopbits are escape none
is local echo is    : 8 :
noinit is noreset is 1
hangup is          : C-a :
                   no :
                   no : no
                   : no
```



```
nolock is      : no send_cmd
is            : sz -vv
receive_cmd is : rz -vv -E
imap is omap   : :
is emap is     : crclrf,delbs, : none
logfile is     : none
initstring     : not set
exit_after is  : no
exit is
```

Type [C-a] [C-h] to see available commands Terminal ready

Optionally, user may overwrite baud rate for experiment.

- Example:

```
# dpu-tty.py -n dpu1 -b 9600
```

Optionally, user may overwrite TTY device for experiment.

- Example:

```
# dpu-tty.py -n dpu2 -t ttyS4
```




2.1.12 CMIS firmware upgrade

2.1.12.1 CMIS firmware version show commands

The `sfputil` command shows the current major and minor versions of active/inactive firmware, running Image details. The output may vary based on the single vs dual bank supported modules.

sfputil show fwversion

- Usage:

```
sfputil show fwversion PORT_NAME
```

- Example:

```
admin@sonic:~$ sfputil show fwversion Ethernet180 Image A
Version: 0.3.5
Image B Version: 0.3.5 Factory Image
Version: 0.0.0 Running Image: A
Committed Image: A Active
Firmware: 0.3.5 Inactive
Firmware: 0.3.5
```

2.1.12.2 CMIS firmware upgrade commands

The `sfputil` commands are used to download/upgrade firmware on transceiver modules. The download/upgrade actually happens using set of CMIS CDB commands. The module may replace the existing image or copy into the inactive bank of the module. The host issues a download complete CDB command when the entire firmware image has been written to LPL or EPL pages. Each step can be verified using the '`sfputil show fwversion PORT_NAME`'

sfputil firmware download

This command is used for downloading firmware to upgrade the transceiver module.

- Usage:

```
sfputil firmware download PORT_NAME FILE_PATH
```

- Example:

```
admin@sonic:~$ sfputil firmware download Ethernet180
AEC_Camano_YCable__0.3.6_20230905.bin
CDB: Starting firmware download
Downloading ... [#####] 100% CDB:
firmware download complete
Firmware download complete success Total
download Time: 0:01:55.731397

admin@sonic:~$ sfputil show fwversion Ethernet180 Image A
Version: 0.3.5
Image B Version: 0.3.6 Factory Image
Version: 0.0.0 Running Image: A
Committed Image: A Active
Firmware: 0.3.5 Inactive
Firmware: 0.3.6
```

sfputil firmware run

This command is used to start and run a downloaded image. This command transfers control from the currently running firmware to a new firmware.



- Usage:

```
sfputil firmware run PORT_NAME
```

- Example:

```
admin@sonic:~$ sfputil firmware run Ethernet180 Running firmware:
Non-hitless Reset to Inactive Image Firmware run in mode=0 success

admin@sonic:~$ sfputil show fwversion Ethernet180 Image A
Version: 0.3.5
Image B Version: 0.3.6 Factory Image
Version: 0.0.0 Running Image: B
Committed Image: A Active
Firmware: 0.3.6 Inactive
Firmware: 0.3.5
```

sfputil firmware commit

This command to commit the running image so that the module will boot from it on future boots. • Usage:

```
sfputil firmware commit PORT_NAME
```

- Example:

```
admin@sonic:~$ sfputil firmware commit Ethernet180 Firmware
commit successful

admin@sonic:~$ sfputil show fwversion Ethernet180 Image A
Version: 0.3.5
Image B Version: 0.3.6 Factory Image
Version: 0.0.0 Running Image: B
Committed Image: B Active
Firmware: 0.3.6 Inactive
Firmware: 0.3.5
```

2.1.12.3 CMIS firmware target mode commands

This command is vendor-specific and supported on the modules to set the target mode to perform remote firmware upgrades. The target modes can be set as 0 (local- E0), 1 (remote end E1), or 2 (remote end E2). Depending on the mode set, the remote or local end will respond to CDB/I2C commands from host's E0 end. After setting the target mode, we can use **sfputil** firmware upgrade commands, will be executed on the module for which target mode is set.

Example of the module supporting target mode

External image: https://github.com/AnoopKamath/sonic-utilities_remote_upgrade/assets/115578705/c3b0bb62-eb14-4b05-b0a8-96b8c082455a



sfputil firmware target

- Usage:

```
sfputil firmware target [OPTIONS] PORT_NAME TARGET
Select target end for firmware download 0-(local)
1-(remote-A)
2-(remote-B)
```

- Example:

```
admin@sonic:~$ sfputil firmware target Ethernet180 1 Target Mode set
to 1
```

2.1.13 CMIS debug

2.1.13.1 CMIS debug loopback

This command is the standard CMIS diagnostic control used for troubleshooting link and performance issues between the host switch and transceiver module.

sfputil debug loopback

- Usage:

```
sfputil debug loopback PORT_NAME LOOPBACK_MODE <enable/disable>

Valid values for loopback mode
host-side-input: host side input loopback mode host-side-output:
host side output loopback mode media-side-input: media side input
loopback mode media-side-output: media side output loopback
mode
```

- Example:

```
admin@sonic:~$ sfputil debug loopback Ethernet88 host-side-input enable admin@sonic:~$ sfputil
debug loopback Ethernet88 media-side-output disable
```

2.1.14 DHCP Relay

2.1.14.1 DHCP Relay show commands

This sub-section of commands is used to show the DHCP Relay IP address(es) in a VLAN interface and show dhcpv6_relay counter of a VLAN.

show dhcp_relay ipv4 helper

This command is used to show ipv4 dhcp_relay helper.

- Usage:

```
show dhcp_relay ipv4 helper
```



- Example:

```
admin@sonic:~$ show dhcp_relay ipv4 helper ----- --
-----
Vlan1000      192.0.0.1
               192.0.0.2
-----      -----
```

show dhcp_relay ipv6 destination

This command is used to show ipv6 dhcp_relay destination. • Usage:

```
show dhcp_relay ipv6 destination
```

- Example:

```
admin@sonic:~$ show dhcp_relay ipv6 destination
-----  -----
Vlan1000  fc02:2000::1
          fc02:2000::2
          fc02:2000::3
          fc02:2000::4
-----  -----
```

show dhcp_relay ipv6 counters

This command is used to show ipv6 dhcp_relay counters.

- Usage:

```
show dhcp_relay ipv6 counters
```

- Example:

```
admin@sonic:~$ sudo sonic-clear dhcp_relay counters
      Message Type ---      Vlan1000 --
-----
          Unknown          0
          Solicit          0
          Advertise        0
          Request          5
          Confirm Renew    0
          Rebind Reply     0
          Release Decline  0
          Reconfigure      0
          Information-Request 0
          Relay-Forward    0
          Relay-Reply      0
          Malformed        0
                          0
                          0
                          0
```

2.1.14.2 DHCP Relay clear commands

This sub-section of commands is used to clear the DHCP Relay counters.

sonic-clear dhcp_relay ipv6 counter



This command is used to clear ipv6 dhcp_relay counters.

- Usage:

```
sonic-clear dhcp_relay ipv6 counter [-i <interface>]
```

- Example:

```
admin@sonic:~$ sudo sonic-clear dhcp_relay ipv6 counters
```

2.1.14.3 DHCP Relay config commands

This sub-section of commands is used to add or remove the DHCP Relay Destination IP address(es) for a VLAN interface.

config vlan dhcp_relay add

This command is used to add a DHCP Relay Destination IP address or multiple IP addresses to a VLAN. Note that more than one DHCP Relay Destination IP address can be added on a VLAN interface.

- Usage:

```
config vlan dhcp_relay add <vlan_id> <dhcp_relay_destination_ips>
```

- Example:

```
admin@sonic:~$ sudo config vlan dhcp_relay add 1000 7.7.7.7 Added DHCP relay
destination address ['7.7.7.7'] to Vlan1000 Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config vlan dhcp_relay add 1000 7.7.7.7 1.1.1.1 Added DHCP relay
destination address ['7.7.7.7', '1.1.1.1'] to Vlan1000 Restarting DHCP relay service...
```

config vlan dhcp_relay delete

This command is used to delete a configured DHCP Relay Destination IP address or multiple IP addresses from a VLAN interface.

- Usage:

```
config vlan dhcp_relay del <vlan-id> <dhcp_relay_destination_ips>
```

- Example:

```
admin@sonic:~$ sudo config vlan dhcp_relay del 1000 7.7.7.7 Removed DHCP
relay destination address 7.7.7.7 from Vlan1000 Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config vlan dhcp_relay del 1000 7.7.7.7 1.1.1.1
Removed DHCP relay destination address ('7.7.7.7', '1.1.1.1') from Vlan1000 Restarting DHCP relay
service...
```



config dhcp_relay ipv4 helper add/del

This command is used to add or delete IPv4 DHCP Relay helper addresses to a VLAN. Note that more than one DHCP Relay helper addresses can be operated on a VLAN interface.

- Usage:

```
config dhcp_relay ipv4 helper (add | del) <vlan_id> <dhcp_helper_ips>
```

- Example:

```
admin@sonic:~$ sudo config dhcp_relay ipv4 helper add 1000 7.7.7.7 Added DHCP relay
address [7.7.7.7] to Vlan1000
Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv4 helper add 1000 7.7.7.7 1.1.1.1 Added DHCP relay
address [7.7.7.7, 1.1.1.1] to Vlan1000
Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv4 helper del 1000 7.7.7.7 Removed DHCP relay
address [7.7.7.7] from Vlan1000
Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv4 helper del 1000 7.7.7.7 1.1.1.1 Removed DHCP relay
address [7.7.7.7, 1.1.1.1] from Vlan1000
Restarting DHCP relay service...
```

config dhcp_relay ipv6 destination add/del

This command is used to add or del IPv6 DHCP Relay destination addresses to a VLAN. Note that more than one DHCP Relay Destination addresses can be operated on a VLAN interface.

- Usage:

```
config dhcp_relay ipv6 destination (add | del) <vlan_id> <dhcp_destination_ips>
```

- Example:

```
admin@sonic:~$ sudo config dhcp_relay ipv6 destination add 1000 fc02:2000::1 Added DHCP relay
address [fc02:2000::1] to Vlan1000
Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv6 destination add 1000 fc02:2000::1 fc02:2000::2
Added DHCP relay address [fc02:2000::1, fc02:2000::2] to Vlan1000 Restarting DHCP
relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv6 destination del 1000 fc02:2000::1 Removed DHCP relay
address [fc02:2000::1] from Vlan1000
Restarting DHCP relay service...
```

```
admin@sonic:~$ sudo config dhcp_relay ipv6 destination del 1000 fc02:2000::1 fc02:2000::2
```





Removed DHCP relay address [fc02:2000::1, fc02:2000::2] from Vlan1000 Restarting DHCP relay service...

[Go Back To Beginning of the document or Beginning of this section](#)

2.1.15 Drop Counters

This section explains all the Configurable Drop Counters show commands and configuration options that are supported in SONiC.

2.1.15.1 Drop Counters show commands

show dropcounters capabilities

This command is used to show the drop counter capabilities that are available on this device. It displays the total number of drop counters that can be configured on this device as well as the drop reasons that can be configured for the counters.

- Usage:

```
show dropcounters capabilities
```

- Examples:

```
admin@sonic:~$ show dropcounters capabilities
```

Counter Type	Total --
-----	----
PORT_INGRESS_DROPS	3
SWITCH_EGRESS_DROPS	2

```
PORT_INGRESS_DROPS: L2_ANY
                    SMAC_MULTICAST
                    SMAC_EQUALS_DMACH
                    INGRESS_VLAN_FILTER
                    EXCEEDS_L2_MTU
                    SIP_CLASS_E
                    SIP_LINK_LOCAL
                    DIP_LINK_LOCAL
                    UNRESOLVED_NEXT_HOP
                    DECAP_ERROR
```

```
SWITCH_EGRESS_DROPS:
                    L2_ANY
                    L3_ANY
                    A_CUSTOM_REASON
```

show dropcounters configuration

This command is used to show the current running configuration of the drop counters on this device. • Usage:

```
show dropcounters configuration [-g <group name>]
```

- Examples:



```
admin@sonic:~$ show dropcounters configuration
```

Counter	Alias	Group	Type	Reasons	Description
DEBUG_0	RX_LEGIT	LEGIT	PORT_INGRESS_DROPS	SMAC_EQUALS_DMAC	Legitimate port-level RX pipeline drops
DEBUG_1	TX_LEGIT	None	SWITCH_EGRESS_DROPS	EGRESS_VLAN_FILTER	Legitimate switch-level TX pipeline drops

```
admin@sonic:~$ show dropcounters configuration -g LEGIT
```

Counter	Alias	Group	Type	Reasons	Description
DEBUG_0	RX_LEGIT	LEGIT	PORT_INGRESS_DROPS	SMAC_EQUALS_DMAC	Legitimate port-level RX pipeline drops
				INGRESS_VLAN_FILTER	

show dropcounters counts

This command is used to show the current statistics for the configured drop counters. Standard drop counters are displayed as well for convenience.

Because clear (see below) is handled on a per-user basis different users may see different drop counts.

- Usage:

```
show dropcounters counts [-g <group name>] [-t <counter type>]
```

- Example:

```
admin@sonic:~$ show dropcounters counts
```

IFACE -	STATE	RX_ERR	RX_DROPS	TX_ERR -	TX_DROPS	RX_LEGIT --
Ethernet0	U	10	100	0	0	20
Ethernet4	U	0	1000	0	0	100
Ethernet8	U	100	10	0	0	0

```
DEVICE TX_LEGIT -
```

```
-----sonic
1000
```

```
admin@sonic:~$ show dropcounters counts -g LEGIT
```

IFACE -	STATE	RX_ERR	RX_DROPS	TX_ERR -	TX_DROPS	RX_LEGIT --
Ethernet0	U	10	100	0	0	20
Ethernet4	U	0	1000	0	0	100
Ethernet8	U	100	10	0	0	0

```
admin@sonic:~$ show dropcounters counts -t SWITCH_EGRESS_DROPS DEVICE
```

```
TX_LEGIT
-----sonic
1000
```



2.1.15.2 Drop Counters config commands

config dropcounters install

This command is used to initialize a new drop counter. The user must specify a name, type, and initial list of drop reasons.

This command will fail if the given name is already in use, if the type of counter is not supported, or if any of the specified drop reasons are not supported. It will also fail if all available counters are already in use on the device.

- Usage:

```
config dropcounters install <counter name> <counter type> <reasons list> [-d <description>] [-g <group>] [-a <alias>]
```

- Example:

```
admin@sonic:~$ sudo config dropcounters install DEBUG_2 PORT_INGRESS_DROPS  
[EXCEEDS_L2_MTU,DECAP_ERROR] -d "More port ingress drops" -g BAD -a BAD_DROPS
```

config dropcounters add_reasons

This command is used to add drop reasons to an already initialized counter.

This command will fail if any of the specified drop reasons are not supported. • Usage:

```
config dropcounters add_reasons <counter name> <reasons list>
```

- Example:

```
admin@sonic:~$ sudo config dropcounters add_reasons DEBUG_2 [SIP_CLASS_E]
```

config dropcounters remove_reasons

This command is used to remove drop reasons from an already initialized counter. • Usage:

```
config dropcounters remove_reasons <counter name> <reasons list>
```

- Example:

```
admin@sonic:~$ sudo config dropcounters remove_reasons DEBUG_2 [SIP_CLASS_E]
```

config dropcounters delete

This command is used to delete a drop counter. • Usage:

```
config dropcounters delete <counter name>
```

- Example:

```
admin@sonic:~$ sudo config dropcounters delete DEBUG_2
```



2.1.15.3 Drop Counters clear commands

sonic-clear dropcounters

This command is used to clear drop counters. This is done on a per-user basis.

- Usage:

```
sonic-clear dropcounters
```

- Example:

```
admin@sonic:~$ sonic-clear dropcounters Cleared
drop counters
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.16 Dynamic Buffer Management

This section explains all the show and configuration commands regarding the dynamic buffer management.

Dynamic buffer management is responsible for calculating buffer size according to the ports' configured speed and administrative state. In order to enable dynamic buffer management feature, the ports' speed must be configured. For this please refer Interface naming mode config commands

2.1.16.1 Configuration commands

configure shared headroom pool

This command is used to configure the shared headroom pool. The shared headroom pool can be enabled in the following ways:

- Configure the over subscribe ratio. In this case, the size of shared headroom pool is calculated as the accumulative xoff of all of the lossless PG divided by the over subscribe ratio.
- Configure the size.

In case both of the above parameters have been configured, the `size` will take effect. To disable shared headroom pool, configure both parameters to zero.

- Usage:

```
config buffer shared-headroom-pool over-subscribe-ratio <over-subscribe-ratio> config buffer shared-headroom-pool size <size>
```

The range of over-subscribe-ratio is from 1 to number of ports inclusive. • Example:

```
admin@sonic:~$ sudo config shared-headroom-pool over-subscribe-ratio 2 admin@sonic:~$
sudo config shared-headroom-pool size 1024000
```

configure a lossless buffer profile

This command is used to configure a lossless buffer profile.

- Usage:

```
config buffer profile add <profile_name> --xon <xon_threshold> --xoff <xoff_threshold> [-size
<size>] [-dynamic_th <dynamic_th_value>] [-pool <ingress_lossless_pool_name>]
config buffer profile set <profile_name> --xon <xon_threshold> --xoff <xoff_threshold> [-size
<size>] [-dynamic_th <dynamic_th_value>] [-pool <ingress_lossless_pool_name>]
config buffer profile remove <profile_name>
```



All the parameters are divided to two groups, one for headroom and one for dynamic_th. For any command at least one group of parameters should be provided. For headroom parameters:

- xon is mandatory.
- If shared headroom pool is disabled:
 - At least one of xoff and size should be provided and the other will be optional and conducted via the formula $xon + xoff = size$.
 - $xon + xoff \leq size$; For Mellanox platform $xon + xoff == size$

shared headroom pool is enabled:

- xoff should be provided.
- $size = xoff$ if it is not provided.

If only headroom parameters are provided, the dynamic_th will be taken from CONFIG_DB.DEFAULT_LOSSLESS_BUFFER_PARAMETER.default_dynamic_th.

If only dynamic_th parameter is provided, the headroom_type will be set as dynamic and xon, xoff and size won't be set. This is only used for non default dynamic_th. In this case, the profile won't be deployed to ASIC directly. It can be configured to a lossless PG and then a dynamic profile will be generated based on the port's speed, cable length, and MTU and deployed to the ASIC.

The subcommand add is designed for adding a new buffer profile to the system.

The subcommand set is designed for modifying an existing buffer profile in the system. For a profile with dynamically calculated headroom information, only dynamic_th can be modified.

The subcommand remove is designed for removing an existing buffer profile from the system. When removing a profile, it shouldn't be referenced by any entry in CONFIG_DB.BUFFER_PG.

- Example:

```
admin@sonic:~$ sudo config buffer profile add profile1 --xon 18432 --xoff 18432
admin@sonic:~$ sudo config buffer profile remove profile1
```

config interface cable_length

This command is used to configure the length of the cable connected to a port. The cable_length is in unit of meters and must be suffixed with "m".

- Usage:

```
config interface cable_length <interface_name> <cable_length>
```

- Example:

```
admin@sonic:~$ sudo config interface cable_length Ethernet0 40m
```



config interface buffer priority-group lossless

This command is used to configure the priority groups on which lossless traffic

runs. • Usage:

```
config interface buffer priority-group lossless add <interface_name> <pg_map> [profile]
config interface buffer priority-group lossless set <interface_name> <pg_map> [profile]
config interface buffer priority-group lossless remove <interface_name> [<pg_map>]
```

The <pg_map> can be in one of the following two forms:

- For a range of priorities, the lower bound and upper bound connected by a dash, like 3-4
- For a single priority, the number, like 6

The pg-map represents the map of priorities for lossless traffic. It should be a string and in form of a bit map like 3-4. The - connects the lower bound and upper bound of a range of priorities.

The subcommand add is designed for adding a new lossless PG on top of current PGs. The new PG range must be disjoint with all existing PGs.

For example, currently the PG range 3-4 exist on port Ethernet4, to add PG range 4-5 will fail because it isn't disjoint with 3-4. To add PG range 5-6 will succeed. After that both range 3-4 and 5-6 will work as lossless PG.

The override-profile parameter is optional. When provided, it represents the predefined buffer profile for headroom override.

The subcommand set is designed for modifying an existing PG from dynamic calculation to headroom override or vice versa. The pg-map must be an existing PG.

The subcommand remove is designed for removing an existing PG. The option pg-map must be an existing PG. All lossless PGs will be removed in case no pg-map provided.

• Example:

To configure lossless_pg on a port:

```
admin@sonic:~$ sudo config interface buffer priority-group lossless add Ethernet0 3-4
```

To change the profile used for lossless_pg on a port:

```
admin@sonic:~$ sudo config interface buffer priority-group lossless set Ethernet0 3-4 new-profile
```

To remove one lossless priority from a port:

```
admin@sonic:~$ sudo config interface buffer priority-group lossless remove Ethernet0 6
```

To remove all lossless priorities from a port:

```
admin@sonic:~$ sudo config interface buffer priority-group lossless remove Ethernet0
```



config interface buffer queue

This command is used to configure the buffer profiles for queues.

- Usage:

```
config interface buffer queue add <interface_name> <queue_map> <profile> config interface buffer
queue set <interface_name> <queue_map> <profile> config interface buffer queue remove
<interface_name> <queue_map>
```

The <queue_map> represents the map of queues. It can be in one of the following two forms:

- For a range of priorities, the lower bound and upper bound connected by a dash, like 3-4 ▸
- For a single priority, the number, like 6

The subcommand `add` is designed for adding a buffer profile for a group of queues. The new queue range must be disjoint with all queues with buffer profile configured.

For example, currently the buffer profile configured on queue 3-4 on port Ethernet4, to configure buffer profile on queue 4-5 will fail because it isn't disjoint with 3-4. To configure it on range 5-6 will succeed.

The `profile` parameter represents a predefined egress buffer profile to be configured on the queues.

The subcommand `set` is designed for modifying an existing group of queues.

The subcommand `remove` is designed for removing buffer profile on an existing group of queues. •
Example:

To configure buffer profiles for queues on a port:

```
admin@sonic:~$ sudo config interface buffer queue add Ethernet0 3-4
egress_lossless_profile
```

To change the profile used for queues on a port:

```
admin@sonic:~$ sudo config interface buffer queue set Ethernet0 3-4 new-profile
```

To remove a group of queues from a port:

```
admin@sonic:~$ sudo config interface buffer queue remove Ethernet0 3-4
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.16.2 Show commands

show buffer information

This command is used to display the status of buffer pools and profiles currently deployed to the ASIC.

- Usage:

```
show buffer information
```

- Example:





```

admin@sonic:~$ show buffer information Pool:
ingress_lossless_pool
---- -----type
ingress mode
dynamic size
17170432 ---- ----
----
Pool: egress_lossless_pool ---- ----
----
type egress mode
dynamic size
34340822 ---- ----
----
Pool: ingress_lossy_pool ---- ---
----
type ingress mode
dynamic size
17170432 ---- ----
----
Pool: egress_lossy_pool ---- ---
----
type egress mode
dynamic size
17170432 ---- ----
----
Profile: pg_lossless_100000_5m_profile
-----xon -----18432
dynamic_th 0 18432
xoff [BUFFER_POOL:ingress_lossless_pool] 36864
pool -----
size
-----

Profile: q_lossy_profile
----- -----3
dynamic_th [BUFFER_POOL:egress_lossy_pool] 0
pool -----
size
-----

Profile: egress_lossy_profile
----- -----3
dynamic_th [BUFFER_POOL:egress_lossy_pool] 4096
pool -----
size
-----

Profile: egress_lossless_profile
----- -----
dynamic_th 7

```




```

pool      [BUFFER_POOL:egress_lossless_pool] 0
size      -----
-----

Profile: ingress_lossless_profile
-----
dynamic_th [BUFFER_POOL:ingress_lossless_pool] 0
pool      -----
size      -----
-----

Profile: pg_lossless_100000_79m_profile
-----xon -----18432
dynamic_th 0 60416
xoff        [BUFFER_POOL:ingress_lossless_pool] 78848
pool      -----
size      -----
-----

Profile: pg_lossless_100000_40m_profile
-----xon -----18432
dynamic_th 0 38912
xoff        [BUFFER_POOL:ingress_lossless_pool] 57344
pool      -----
size      -----
-----

Profile: ingress_lossy_profile
-----
dynamic_th [BUFFER_POOL:ingress_lossy_pool] 0
pool      -----
size      -----
-----

```

show buffer configuration

This command is used to display the status of buffer pools and profiles currently configured.

- Usage:

```
show buffer configuration
```

- Example:

```

admin@sonic:~$ show buffer configuration Lossless
traffic pattern:
-----
default_dynamic_th      0
over_subscribe_ratio 0 -----
-----
Pool: ingress_lossless_pool ----
-----

```





```

type ingress mode
dynamic ---- -----
---
Pool: egress_lossless_pool ---- ----
----
type egress mode
dynamic size
34340822 ---- ----
----
Pool: ingress_lossy_pool ---- ---
-----
type ingress mode
dynamic ---- -----
---
Pool: egress_lossy_pool ---- ---
-----
type egress mode
dynamic ---- -----
---
Profile: q_lossy_profile
-----3
dynamic_th [BUFFER_POOL:egress_lossy_pool] 0
pool -----
size
-----

Profile: egress_lossy_profile
-----3
dynamic_th [BUFFER_POOL:egress_lossy_pool] 4096
pool -----
size
-----

Profile: egress_lossless_profile
-----7
dynamic_th [BUFFER_POOL:egress_lossless_pool] 0
pool -----
size
-----

Profile: ingress_lossless_profile
-----0
dynamic_th [BUFFER_POOL:ingress_lossless_pool] 0
pool -----
size
-----

Profile: ingress_lossy_profile
-----3
dynamic_th [BUFFER_POOL:ingress_lossy_pool]
pool

```





```
size      0
-----
```

Go Back To Beginning of the document or Beginning of this section

2.1.17 Feature

SONiC includes a capability in which Feature state can be enabled/disabled which will make corresponding feature docker container to start/stop.

Also SONiC provide capability in which Feature docker container can be automatically shut down and restarted if one of critical processes running in the container exits unexpectedly. Restarting the entire feature container ensures that configuration is reloaded and all processes in the feature container get restarted, thus increasing the likelihood of entering a healthy state.

2.1.17.1 Feature show commands

show feature config

Shows the config of given feature or all if no feature is given. The “fallback” is shown only if configured. The fallback defaults to “true” when not configured.

- Usage:

```
show feature config [<feature name>]
```

- Example:

```
admin@sonic:~$ show feature config
```

Feature	State	AutoRestart	Owner	fallback
-----bgp	-----	-----	-----	-----
database	enabled	enabled disabled	local local	
dhcp_relay	enabled	enabled enabled	kube kube	
lldp	enabled	enabled enabled	local local	
mgmt-framework	enabled	enabled enabled	kube kube	true
nat	enabled	enabled enabled	local kube	
pmon	disabled	enabled enabled	local local	
radv	enabled	enabled	local	
sflow	enabled	enabled	kube	
snmp	disabled			
swss	enabled			
syncd	enabled			
teamd	enabled			
telemetry	enabled			
	enabled			

show feature status

Shows the status of given feature or all if no feature is given. The “fallback” defaults to “true” when not configured. The subset of features are configurable for remote management and only those report additional data.

- Usage:

```
show feature status [<feature name>]
```

- Example:

```
admin@sonic:~$ show feature status
```





config feature owner

Configures the owner for a feature as “local” or “kube”. The “local” implies starting the feature container from local image. The “kube” implies that kubernetes server is made eligible to deploy the feature. The deployment of a feature by kubernetes is conditional based on many factors like, whether the kube server is configured or not, connected-to-kube-server or not and if that master has manifest for this feature for this switch or not and more. At some point in future, the deployment *could* happen and till that point the feature can run from local image, called “fallback”. The fallback is allowed by default and it could be toggled to “not allowed”. When fallback is not allowed, the feature would run only upon deployment by kubernetes master.

- Usage:

```
config feature owner [<feature name>] [local/kube]
```

- Example:

```
admin@sonic:~$ sudo config feature owner snmp kube
```

config feature fallback

Features configured for “kube” deployment could be allowed to fallback to using local image, until the point of successful kube deployment. The fallback is allowed by default.

- Usage:

```
config feature fallback [<feature name>] [on/off]
```

- Example:

```
admin@sonic:~$ sudo config feature fallback snmp on
```

show feature autorestart

This command will display the status of auto-restart for feature container.

- Usage:

```
show feature autorestart [<feature_name>]
```

```
admin@sonic:~$ show feature autorestart
```

Feature	AutoRestart
-----bgp	-----
database	enabled
dhcp_relay	always_enabled
lldp	enabled enabled
pmon	enabled enabled
radv	enabled enabled
snmp	enabled enabled
swss	enabled
syncd	
teamd	
telemetry	

Optionally, you can specify a feature name in order to display status for that feature



2.1.17.2 Feature config commands

config feature state <feature_name>

This command will configure the state for a specific feature.

- Usage:

```
config feature state <feature_name> (enabled | disabled) admin@sonic:~$
sudo config feature state bgp disabled
```

To make the command wait until the corresponding feature container stops(starts) use `--block` options:

- Usage:

```
admin@sonic:~$ config feature state bgp enabled --block
```

config feature autorestart <feature_name> <autorestart_status>

This command will configure the status of auto-restart for a specific feature container.

- Usage:

config feature autorestart <feature_name> <autorestart_status>

This command will configure the status of auto-restart for a specific feature container.

- Usage:

```
config feature autorestart <feature_name> (enabled | disabled) admin@sonic:~$ sudo
config feature autorestart bgp disabled
```

NOTE: If the existing state or auto-restart value for a feature is “always_enabled” then config commands are don’t care and will not update state/auto-restart value.

Go Back To Beginning of the document or Beginning of this section

2.1.18 Flow Counters

This section explains all the Flow Counters show commands, clear commands and config commands that are supported in SONiC. Flow counters are usually used for debugging, troubleshooting and performance enhancement processes. Flow counters supports case like:

- Host interface traps (number of received traps per Trap ID)
- Routes matching the configured prefix pattern (number of hits and number of bytes)

2.1.18.1 Flow Counters show commands

show flowcnt-trap stats

This command is used to show the current statistics for the registered host interface traps.

Because clear (see below) is handled on a per-user basis different users may see different counts.

- Usage:

```
show flowcnt-trap stats
```

- Example:

```
admin@sonic:~$ show flowcnt-trap stats
Trap Name Packets Bytes PPS
-----
dhcp 100 2,000 50.25/s
For multi-ASIC:
admin@sonic:~$ show flowcnt-trap stats
ASIC ID Trap Name Packets Bytes PPS
-----
asic0 dhcp 100 2,000 50.25/s
asic1 dhcp 200 3,000 45.25/s
```






show flowcnt-route stats

This command is used to show the current statistics for route flow patterns.

Because clear (see below) is handled on a per-user basis different users may see different counts.

- Usage:

```
show flowcnt-route stats
show flowcnt-route stats pattern <route_pattern> [--vrf <vrf>] show flowcnt-route
stats route <route_prefix> [--vrf <vrf>]
```

- Example:

```
admin@sonic:~$ show flowcnt-route stats
Route pattern          VRF          Matched routes          Packets
Bytes
-----
3.3.0.0/16             default       3.3.1.0/24              100
4543
                      3.3.2.3/32              3443
929229
                      3.3.0.0/16              0
0 2000::/64            default       2000::1/128             100
4543
```

The “pattern” subcommand is used to display the route flow counter statistics by route pattern.

- Example:

```
admin@sonic:~$ show flowcnt-route stats pattern 3.3.0.0/16
Route pattern          VRF          Matched routes          Packets
Bytes
-----
3.3.0.0/16             default       3.3.1.0/24              100
4543
                      3.3.2.3/32              3443
929229
                      3.3.0.0/16              0
0
```

The “route” subcommand is used to display the route flow counter statistics by route prefix.

```
admin@sonic:~$ show flowcnt-route stats route 3.3.3.2/32 --vrf Vrf_1
Route      VRF      Route      Pattern  Packets Bytes
3.3.3.2/32 Vrf_1    3.3.0.0/16 100      4543
```

2.1.18.2 Flow Counters clear commands

sonic-clear flowcnt-trap

This command is used to clear the current statistics for the registered host interface traps. This is done on a per-user basis.

- Usage:

```
sonic-clear flowcnt-trap
```

- Example:

```
admin@sonic:~$ sonic-clear flowcnt-trap
Trap Flow Counters were successfully cleared
```





sonic-clear flowcnt-route

This command is used to clear the current statistics for the route flow counter. This is done on a per-user basis.

- Usage:

```
sonic-clear flowcnt-route
sonic-clear flowcnt-route pattern <route_pattern> [--vrf <vrf>] sonic-clear flowcnt-
route route <prefix> [--vrf <vrf>]
```

- Example:

```
admin@sonic:~$ sonic-clear flowcnt-route Route Flow
Counters were successfully cleared
```

The “pattern” subcommand is used to clear the route flow counter statistics by route pattern.

- Example:

```
admin@sonic:~$ sonic-clear flowcnt-route pattern 3.3.0.0/16 --vrf Vrf_1 Flow Counters of all
routes matching the configured route pattern were successfully cleared
```

The “route” subcommand is used to clear the route flow counter statistics by route prefix. • Example:

```
admin@sonic:~$ sonic-clear flowcnt-route route 3.3.3.2/32 --vrf Vrf_1 Flow Counters of the
specified route were successfully cleared
```

2.1.18.3 Flow Counters config commands

config flowcnt-route pattern add

This command is used to add or update the route pattern which is used by route flow counter to match route entries.

- Usage:

```
config flowcnt-route pattern add <prefix> [--vrf <vrf>] [--max <max_match_count>]
```

- Example:

```
admin@sonic:~$ config flowcnt-route pattern add 2.2.0.0/16 --vrf Vrf_1 --max 50
```

config flowcnt-route pattern remove

This command is used to remove the route pattern which is used by route flow counter to match route entries.

- Usage:

```
config flowcnt-route pattern remove <prefix> [--vrf <vrf>]
```

- Example:

```
admin@sonic:~$ config flowcnt-route pattern remove 2.2.0.0/16 --vrf Vrf_1
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)





2.1.19 Gearbox

This section explains all the Gearbox PHY show commands that are supported in SONiC.

2.1.19.1 Gearbox show commands

This sub-section contains the show commands that are supported for gearbox phy.

show gearbox interfaces status

This command displays information about the gearbox phy interface lanes, speeds and status. Data is displayed for both MAC side and line side of the gearbox phy

- Usage:

```
show gearbox interfaces status
```

- Example:

```
home/admin# show gearbox interfaces status
```

show gearbox phys status

This command displays basic information about the gearbox phys configured on the switch.

- Usage:

```
show gearbox phys status
```

- Example:

```
/home/admin# show gearbox phys status
```

[Go Back To Beginning of the document or Beginning of this section](#)

This command is used to change device hostname without traffic being impacted.

- Usage:

```
config hostname <new_hostname>
```

- Example:

```
admin@sonic:~$ sudo config hostname CSW06
```

Please note loaded setting will be lost after system reboot. To preserve setting, run `config save`.





2.1.21 Hash

This section explains the various show and configuration commands available for user.

2.1.21.1 Hash Show Commands

This subsection explains how to display switch hash configuration.

show switch-hash global

This command displays switch hash global configuration.

- Usage:

```
show switch-hash global
```

- Options:

- *-j, -json*: display in JSON format

- Example:

```
admin@sonic:~$ show switch-hash global
```

show switch-hash capabilities

This command displays switch hash capabilities.

- Usage:

```
show switch-hash capabilities
```

- Options:

- *-j, -json*: display in JSON format

- Example:

```
admin@sonic:~$ show switch-hash capabilities
```




2.1.21.2 Hash Config Commands

This subsection explains how to configure switch hash.

config switch-hash global ecmp/lag hash

This command is used to manage switch hash global configuration.

- Usage:

```
config switch-hash global ecmp-hash <hash_field_list> config switch-  
hash global lag-hash <hash_field_list>
```

- Parameters:

- *hash_field_list*: hash fields for hashing packets going through ECMP/LAG

- Examples:

```
admin@sonic:~$ config switch-hash global ecmp-hash \ 'DST_MAC' \  
'SRC_MAC' \  
  
'ETHERTYPE' \  
'IP_PROTOCOL' \  
'DST_IP' \ 'SRC_IP' \  
'L4_DST_PORT' \  
'L4_SRC_PORT' \  
'INNER_DST_MAC' \  
'INNER_SRC_MAC' \  
'INNER_ETHERTYPE' \  
'INNER_IP_PROTOCOL' \  
'INNER_DST_IP' \  
'INNER_SRC_IP' \  
'INNER_L4_DST_PORT' \  
'INNER_L4_SRC_PORT' \  
'IPV6_FLOW_LABEL'  
  
admin@sonic:~$ config switch-hash global lag-hash \ 'DST_MAC' \  
'SRC_MAC' \  
  
'ETHERTYPE' \  
'IP_PROTOCOL' \  
'DST_IP' \ 'SRC_IP' \  
'L4_DST_PORT' \  
'L4_SRC_PORT' \  
'INNER_DST_MAC' \  
'INNER_SRC_MAC' \  
'INNER_ETHERTYPE' \  
'INNER_IP_PROTOCOL' \  
'INNER_DST_IP' \  
'INNER_SRC_IP' \  
'INNER_L4_DST_PORT' \  
'INNER_L4_SRC_PORT' \  
'IPV6_FLOW_LABEL'
```





config switch-hash global ecmp/lag hash algorithm

This command is used to manage switch hash algorithm global configuration.

- Usage:

```
config switch-hash global ecmp-hash-algorithm <hash_algorithm> config switch-  
hash global lag-hash-algorithm <hash_algorithm>
```

- Parameters:

- *hash_algorithm*: hash algorithm for hashing packets going through ECMP/LAG

- Examples:

```
admin@sonic:~$ config switch-hash global ecmp-hash-algorithm 'CRC' admin@sonic:~$  
config switch-hash global lag-hash-algorithm 'CRC'
```

2.1.22 Interfaces

2.1.22.1 Interface Show Commands

This sub-section lists all the possible show commands for the interfaces available in the device. Following example gives the list of possible shows on interfaces. Subsequent pages explain each of these commands in detail.

```
admin@sonic:~$ show interfaces -?  
Show details of the network interfaces  
Options:  
-?, -h, --help Show tis message and exit.
```

show interfaces autoneg

This show command displays the port auto negotiation status for all interfaces i.e. interface name, auto negotiation mode, speed, advertised speeds, interface type, advertised interface types, operational status, admin status. For a single interface, provide the interface name with the sub-command.

- Usage:

```
show interfaces autoneg status  
show interfaces autoneg status <interface_name>
```

- Example:

```
admin@sonic:~$ show interfaces autoneg status
```

Interface	Admin	Auto-Neg Mode	Speed	Adv Speeds	Type	Adv Types	Oper
Ethernet0	up	enabled	25G	10G, 25G	CR	CR, CR4	up
Ethernet4	up	disabled	100G	all	CR4	all	up

- Example:

```
admin@sonic:~$ show interfaces autoneg status Ethernet49
```

Interface	Admin	Auto-Neg Mode	Speed	Adv Speeds	Type	Adv Types	Oper
Ethernet49	up	disabled	100G	N/A	CR4	N/A	up



show interfaces breakout (Versions >= 202006)

This show command displays the port capability for all interfaces i.e. index, lanes, default_brkout_mode, breakout_modes(i.e. available breakout modes) and brkout_mode (i.e. current breakout mode). To display current breakout mode, “current-mode” subcommand can be used. For a single interface, provide the interface name with the sub-command.

- Usage:

```
show interfaces breakout
show interfaces breakout current-mode
show interfaces breakout current-mode <interface_name>
```

- Example:

```
admin@sonic:~$ show interfaces breakout
{
  "Ethernet0": {
    "index": "1,1,1,1", "default_brkout_mode":
    "1x100G[40G]", "child ports": "Ethernet0",
    "child port speed": "100G",
    "breakout_modes": "1x100G[40G],2x50G,4x25G[10G]", "Current
    Breakout Mode": "1x100G[40G]",
    "lanes": "65,66,67,68",
    "alias_at_lanes": "Eth1/1, Eth1/2, Eth1/3, Eth1/4" },... continue
  }
}
```

The “current-mode” subcommand is used to display current breakout mode for all interfaces.

```
admin@lnos-x1-a-fab01:~$ show interfaces breakout current-mode
```

<u>Interface</u>	<u>Current Breakout Mode</u>
Ethernet0	4x25G[10G]
Ethernet8	4x25G[10G]
Ethernet12	4x25G[10G]

```
admin@lnos-x1-a-fab01:~$ show interfaces breakout current-mode Ethernet0
```

<u>Interface</u>	<u>Current Breakout Mode</u>
Ethernet0	4x25G[10G]



show interfaces counters

This show command displays packet counters for all interfaces(except the “show interface detailed” command) since the last time the counters were cleared. To display l3 counters “rif” subcommand can be used. There is no facility to display counters for one specific l2 interface. For l3 interfaces a single interface output mode is present. Optional argument “-a” provides two additional columns - RX-PPS and TX_PPS. Optional argument “-p” specify a period (in seconds) with which to gather counters over. To display the detailed per-interface counters “detailed ” subcommand can be used.

- Usage:

```
show interfaces counters [-a|--printall] [-p|--period <period>]
show interfaces counters errors
show interfaces counters rates
show interfaces counters rif [-p|--period <period>] [-i <interface_name>]
show interfaces counters fec-histogram [-i <interface_name>]
show interfaces counters fec-stats
show interfaces counters detailed <interface_name>
```

- Example:

```
admin@sonic:~$ show interfaces counters
```

IFACE	STATE	RX_OK	RX_BPS	RX_UTIL	RX_ERR	RX_DRP
RX_OVR		TX_OK	TX_BPS	TX_UTIL	TX_ERR	TX_DRP
Ethernet0	U	471,729,839,997	653.87 MB/s	12.77%	0	18,682
0 409,682,385,925		556.84 MB/s	10.88%	0	0 0	
Ethernet4	U	453,838,006,636	632.97 MB/s	12.36%	0	1,636
0 388,299,875,056		529.34 MB/s	10.34%	0	0 0	
Ethernet8	U	549,034,764,539	761.15 MB/s	14.87%	0	18,274
0 457,603,227,659		615.20 MB/s	12.02%	0	0 0	
Ethernet12	U	458,052,204,029	636.84 MB/s	12.44%	0	17,614
0 388,341,776,615		527.37 MB/s	10.30%	0	0 0	
Ethernet16	U	16,679,692,972	13.83 MB/s	0.27%	0	17,605
0 18,206,586,265		17.51 MB/s	0.34%	0	0 0	
Ethernet20	U	47,983,339,172	35.89 MB/s	0.70%	0	2,174
0 58,986,354,359		51.83 MB/s	1.01%	0	0 0	
Ethernet24	U	33,543,533,441	36.59 MB/s	0.71%	0	1,613
0 43,066,076,370		49.92 MB/s	0.97%	0	0 0	

```
admin@sonic:~$ show interfaces counters -i Ethernet4,Ethernet12-16
```



IFACE	STATE	RX_OK	RX_BPS	RX_UTIL	RX_ERR	RX_DRP
RX_OVR	TX_OK	TX_BPS	TX_UTIL	TX_ERR	TX_DRP	TX_OVR
-----	-----	-----	-----	-----	-----	-----
Ethernet4	U	453,838,006,636	632.97 MB/s	12.36%	0	1,636
0 388,299,875,056	529.34 MB/s	10.34%	0	0 0		
Ethernet12	U	458,052,204,029	636.84 MB/s	12.44%	0	17,614
0 388,341,776,615	527.37 MB/s	10.30%	0	0 0		
Ethernet16	U	16,679,692,972	13.83 MB/s	0.27%	0	17,605
0 18,206,586,265	17.51 MB/s	0.34%	0	0 0		

The “errors” subcommand is used to display the interface errors.

- Example:

```
admin@str-s6000-ac-s-11:~$ show interface counters errors
```

IFACE	STATE	RX_ERR	RX_DRP	RX_OVR	TX_ERR	TX_DRP	TX_OVR
-----	-----	-----	-----	-----	-----	-----	-----
Ethernet0	U	0	4	0	0	0	0
Ethernet4	U	0	0	0	0	0	0
Ethernet8	U	0	1	0	0	0	0
Ethernet12	U	0	0	0	0	0	0

The “rates” subcommand is used to display only the interface rates.

- Example:

```
admin@str-s6000-ac-s-11:/usr/bin$ show int counters rates
```

IFACE	STATE	RX_OK	RX_BPS	RX_PPS	RX_UTIL	TX_OK	TX_BPS
TX_PPS	TX_UTIL						
-----	-----	-----	-----	-----	-----	-----	-----
Ethernet0	U	467510	N/A	N/A	N/A	466488	N/A
N/A N/A							
Ethernet4	U	469679	N/A	N/A	N/A	469245	N/A
N/A N/A							
Ethernet8	U	466660	N/A	N/A	N/A	465982	N/A
N/A N/A							
Ethernet12	U	466579	N/A	N/A	N/A	466318	N/A
N/A N/A							

The “rif” subcommand is used to display L3 interface counters. Layer 3 interfaces include router interfaces, portchannels and vlan interfaces.

- Example:

```
admin@sonic:~$ show interfaces counters rif
```

IFACE	RX_OK	RX_BPS	RX_PPS	RX_ERR	TX_OK	TX_BPS
TX_PPS	TX_ERR					
-----	-----	-----	-----	-----	-----	-----



PortChannel0001	62,668	107.81 B/s	1.34/s	3	6	0.02 B/s
0.00/s	0					
PortChannel0002	62,645	107.77 B/s	1.34/s	3	2	0.01 B/s
0.00/s	0					
PortChannel0003	62,481	107.56 B/s	1.34/s	3	3	0.01 B/s
0.00/s	0					
PortChannel0004	62,732	107.88 B/s	1.34/s	2	3	0.01 B/s
0.00/s	0					
Vlan1000	0	0.00 B/s	0.00/s	0	0	0.00 B/s
0.00/s	0					

Optionally, you can specify a layer 3 interface name to display the counters in single interface mode.

- Example:

```
admin@sonic:~$ show interfaces counters rif PortChannel0001 PortChannel0001
```

RX:

3269 packets
778494 bytes
3 error packets 292
error bytes

TX:

0 packets 0
bytes
0 error packets 0
error bytes

Optionally, you can specify a period (in seconds) with which to gather counters over. Note that this function will take `<period>` seconds to execute.

- Example:

```
admin@sonic:~$ show interfaces counters -p 5
```

RX_OVR	IFACE	STATE	RX_OK	RX_BPS	RX_UTIL	RX_ERR	RX_DRP
TX_OK	TX_BPS	TX_UTIL	TX_ERR	TX_DRP	TX_OVR		
-----	-----	-----	-----	-----	-----	-----	-----
0	Ethernet0	U	515	59.14 KB/s	0.00%	0	0
1,305	127.60 KB/s	0.00%	0	0	0		
0	Ethernet4	U	305	26.54 KB/s	0.00%	0	0
279	39.12 KB/s	0.00%	0	0	0		
0	Ethernet8	U	437	42.96 KB/s	0.00%	0	0
182	18.37 KB/s	0.00%	0	0	0		
0	Ethernet12	U	284	40.79 KB/s	0.00%	0	0
160	13.03 KB/s	0.00%	0	0	0		
0	Ethernet16	U	377	32.64 KB/s	0.00%	0	0
214	18.01 KB/s	0.00%	0	0	0		
0	Ethernet20	U	284	36.81 KB/s	0.00%	0	0
138	8758.25 B/s	0.00%	0	0	0		
0	Ethernet24	U	173	16.09 KB/s	0.00%	0	0
169	11.39 KB/s	0.00%	0	0	0		



The “detailed” subcommand is used to display more detailed interface counters. Along with tx/rx counters, it also displays the WRED drop counters that are supported on the platform.

- Example:

```
admin@sonic:~$ show interfaces counters detailed Ethernet8
Packets Received 64 Octets..... 0
Packets Received 65-127 Octets..... 0
Packets Received 128-255 Octets..... 0
Packets Received 256-511 Octets..... 0
Packets Received 512-1023 Octets..... 0
Packets Received 1024-1518 Octets..... 0
Packets Received 1519-2047 Octets..... 0
Packets Received 2048-4095 Octets..... 0
Packets Received 4096-9216 Octets..... 0
Packets Received 9217-16383 Octets..... 0

Total Packets Received Without Errors..... 0
Unicast Packets Received..... 0
Multicast Packets Received..... 0
Broadcast Packets Received..... 0

Jabbers Received..... N/A
Fragments Received..... N/A
Undersize Received..... 0
Overruns Received..... 0

Packets Transmitted 64 Octets..... 0
Packets Transmitted 65-127 Octets..... 0
Packets Transmitted 128-255 Octets..... 0
Packets Transmitted 256-511 Octets..... 0
Packets Transmitted 512-1023 Octets..... 0
Packets Transmitted 1024-1518 Octets..... 0
Packets Transmitted 1519-2047 Octets..... 0
Packets Transmitted 2048-4095 Octets..... 0
Packets Transmitted 4096-9216 Octets..... 0
Packets Transmitted 9217-16383 Octets..... 0

Total Packets Transmitted Successfully..... 0
Unicast Packets Transmitted..... 0
Multicast Packets Transmitted..... 0
Broadcast Packets Transmitted..... 0

WRED Green Dropped Packets..... 0
WRED Yellow Dropped Packets..... 0
WRED Red Dropped Packets..... 0
WRED Total Dropped Packets..... 0

Time Since Counters Last Cleared..... None
```

- NOTE: Interface counters can be cleared by the user with the following command:



```
admin@sonic:~$ sonic-clear counters
```

- NOTE: Layer 3 interface counters can be cleared by the user with the following command:

```
admin@sonic:~$ sonic-clear rifcounters
```

The “fec-histogram” subcommand is used to display the fec histogram for the port.

When data is transmitted, it’s broken down into units called codewords. FEC algorithms add extra data to each codeword that can be used to detect and correct errors in transmission. In a FEC histogram, “bins” represent ranges of errors or specific categories of errors. For instance, Bin0 might represent codewords with no errors, while Bin1 could represent codewords with a single bit error, and so on. The histogram shows how many codewords fell into each bin. A high number in the higher bins might indicate a problem with the transmission link, such as signal degradation.

- Example:

```
admin@str-s6000-ac-s-11:/usr/bin$ show interface counters fec-histogram -i <PORT>
Symbol Errors Per Codeword ----- Codewords --
-----BIN0:
BIN1: 1000000
BIN2: 900000
BIN3: 800000
BIN4: 700000
BIN5: 600000
BIN6: 500000
BIN7: 400000
BIN8: 300000
BIN9: 0
BIN10: 0
BIN11: 0
BIN12: 0
BIN13: 0
BIN14: 0
BIN15: 0
```

The “fec-stats” subcommand is used to display the interface fec related statistic.

- Example:

```
admin@ctd615:~$ show interfaces counters fec-stats
      IFACE      STATE  FEC_CORR  FEC_UNCORR  FEC_SYMBOL_ERR  FEC_PRE_BER
FEC_POST_BER
-----
Ethernet0      U        0          0          0      1.48e-20
0.00e+00
Ethernet8      U        0          0          0      1.98e-19
0.00e+00
Ethernet16     U        0          0          0      1.77e-20
0.00e+00
```



show interfaces description

This command displays the key fields of the interfaces such as Operational Status, Administrative Status, Alias and Description.

- Usage:

```
show interfaces description [<interface_name>]
```

- Example:

```
admin@sonic:~$ show interfaces description
```

Interface	Oper	Admin	Alias	Description
-----	-----	-----	-----	-----
Ethernet0	down	up	hundredGigE1/1 up	T0-1:hundredGigE1/30 T0-
Ethernet4	down		hundredGigE1/2 down	2:hundredGigE1/30
Ethernet8	down		hundredGigE1/3	hundredGigE1/3
Ethernet12	down	down	hundredGigE1/4	hundredGigE1/4

- Example (to only display the description for interface Ethernet4):

```
admin@sonic:~$ show interfaces description Ethernet4
```

Interface	Oper	Admin	Alias	Description
-----	-----	-----	-----	-----
Ethernet4	down	up	hundredGigE1/2 T0-2:hundredGigE1/30	

show interfaces fec status (Versions >= 202311)

This command is to display the FEC status of the selected interfaces. If **interface_name** is not specified, this command shows the FEC status of all interfaces.

- Usage:

```
show interfaces fec status [<interface_name>]
```

- Example:

```
admin@sonic:~$ show interfaces fec status
```

Interface	FEC Oper	FEC Admin
-----	-----	-----
Ethernet0	N/A	rs
Ethernet32	N/A	rs
Ethernet36	N/A	N/A
Ethernet112	N/A	rs
Ethernet116	N/A	rs
Ethernet120	N/A	rs
Ethernet124	rs	auto



show interfaces link-training (Versions >= 202211)

This command is to display the link-training status of the selected interfaces. If **interface_name** is not specified, this command shows the link-training status of all interfaces.

- Usage:

```
show interfaces link-training status [<interface_name>]
```

- Example:

```
admin@sonic:~$ show interfaces link-training status
  Interface      LT Oper    LT Admin   Oper    Admin
  -----
  Ethernet0      trained          on       up       up
  Ethernet8      trained          on       up       up
  Ethernet16      off            off      down     up
  Ethernet24    not trained          on      down     up
  Ethernet32      off            off      down     up
```

- Example (to only display the link-training status of interface Ethernet8):

```
admin@sonic:~$ show interfaces link-training status Ethernet8
  Interface      LT Oper    LT Admin   Oper    Admin
  -----
  Ethernet8      trained          on       up       up
```

show interfaces flap

The show interfaces flap command provides detailed insights into interface events, including the timestamp of the last link down event and the total flap count (number of times the link has gone up and down). This helps in diagnosing stability and connectivity issues.

- Usage:

```
show interfaces flap
```

- Example:

```
admin@sonic:~$ show interfaces flap
Interface      Flap Count    Admin    Oper    Link Down TimeStamp (UTC) Link Up TimeStamp (UTC)
Ethernet0      5             Up       Up       Last flapped : 2024-10-01 10:00:00 (0 days 00:01:23 ago)
Last Link up: 2024-09-30 10:01:03 UTC (1 days 02:30:15 ago)
Ethernet4      Never         Up       Up       Never
Last Link up: 2024-09-30 10:01:03 UTC (1 days 02:30:15 ago)
Ethernet8      1             Up       Up       Last flapped : 2024-10-01 10:01:00 (0 days
00:00:23 ago) Last Link up: 2024-10-02 10:01:03 UTC (5 days 02:30:15 ago)
```

show interfaces errors

The show interface errors command provides detailed statistics and error counters for MAC-level operations on an interface. It displays the status of various operational parameters, error counts, and timestamps for when these errors occurred.

- Usage:

```
show interfaces errors [<interface_name>]
```

- Example:



```
admin@sonic:~$ show interfaces errors Ethernet4
```

Port Errors	Count	Last timestamp(UTC) -----	

oper_error_status	5442	2024-11-02	04:00:05
mac_local_fault	2	2024-11-02	04:00:05
fec_sync_loss	2	2024-11-02	04:00:05
fec_alignment_loss	2	2024-11-02	04:00:05
high_ser_error	2 high	2024-11-02	04:00:05
ber_error	2	2024-11-02	04:00:05
data_unit_crc_error	2	2024-11-02	04:00:05
data_unit_misalignment_error	2	2024-11-02	04:00:05
signal_local_error	2	2024-11-02	04:00:05
mac_remote_fault	2	2024-11-02	04:00:50
crc_rate	2	2024-11-02	04:00:50
data_unit_size	2	2024-11-02	04:00:50
code_group_error	0	Never	
no_rx_reachability	0	Never	

show interfaces mpls

This command is used to display the configured MPLS state for the list of configured interfaces.

- Usage:

```
show interfaces mpls [<interface_name>]
```

- Example:

```
admin@sonic:~$ show interfaces mpls
```

Interface	MPLS State
-----	-----
Ethernet0	disable enable
Ethernet4	enable disable
Ethernet8	disable
Ethernet12	disable
Ethernet16	
Ethernet20	

- Example (to only display the MPLS state for interface Ethernet4):

```
admin@sonic:~$ show interfaces mpls Ethernet4
```

Interface	MPLS State
-----	-----
Ethernet4	enable

show interfaces loopback-action

This command displays the configured loopback action

- Usage:

```
show ip interfaces loopback-action
```



- Example:

```
root@sonic:~# show ip interfaces loopback-action
Interface      Action
-----
Ethernet232    forward
Vlan100
```

show interfaces tpid

This command displays the key fields of the interfaces such as Operational Status, Administrative Status, Alias and TPID.

- Usage:

```
show interfaces tpid [<interface_name>]
```

- Example:

```
admin@sonic:~$ show interfaces tpid
```

Interface ----	Alias	Oper -	Admin	TPID -
Ethernet0	fortyGigE1/1/1	up	up	0x8100 up
Ethernet1	fortyGigE1/1/2	up		0x8100 down
Ethernet2	fortyGigE1/1/3	down		0x8100 down
Ethernet3	fortyGigE1/1/4	down		0x8100 up
Ethernet4	fortyGigE1/1/5	up		0x8100 up
Ethernet5	fortyGigE1/1/6	up		0x8100 up
Ethernet6	fortyGigE1/1/7	up		0x9200 up
Ethernet7	fortyGigE1/1/8	up		0x88A8 up
Ethernet8 ...	fortyGigE1/1/9	up		0x8100
Ethernet63				
PortChannel0001	fortyGigE1/4/16	down	N/A	up 0x8100
PortChannel0002	N/A	up	N/A	up 0x8100
PortChannel0003	up			up 0x8100
PortChannel0004	N/A	up		up 0x8100
admin@sonic:~\$				up 0x8100

- Example (to only display the TPID for interface Ethernet6):

```
admin@sonic:~$ show interfaces tpid Ethernet6
```

<u>Interface</u>	<u>Alias</u>	<u>Oper</u>	<u>Admin</u>	<u>TPID</u>
Ethernet6	fortyGigE1/1/7	up	up	0x9200

```
admin@sonic:~$
```

show interfaces naming_mode

Refer sub-section Interface-Naming-Mode



show interfaces neighbor

This command is used to display the list of expected neighbors for all interfaces (or for a particular interface) that is configured.

- Usage:

```
show interfaces neighbor expected [<interface_name>] -n [<namespace>]
```

- Example:

```
admin@sonic:~$ show interfaces neighbor expected
```

LocalPort	Neighbor	NeighborPort	NeighborLoopback	NeighborMgmt
NeighborType				
Ethernet112	ARISTA01T1	Ethernet1	None	10.16.205.100
ToRRouter				
Ethernet116	ARISTA02T1	Ethernet1	None	10.16.205.101
SpineRouter				
Ethernet120	ARISTA03T1	Ethernet1	None	10.16.205.102
LeafRouter				
Ethernet124	ARISTA04T1	Ethernet1	None	10.16.205.103
LeafRouter				

show interfaces portchannel

This command displays information regarding port-channel interfaces

- Usage:

```
show interfaces portchannel
```

- Example:

```
admin@sonic:~$ show interfaces portchannel
```

Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available, S -selected, D - deselected

No.	Team Dev	Protocol	Ports
-----24		-----	-----
PortChannel24 48	LACP(A)(Up)	Ethernet28(S)	Ethernet24(S)
PortChannel48 40	LACP(A)(Up)	Ethernet52(S)	Ethernet48(S)
PortChannel40	LACP(A)(Up)	Ethernet44(S)	Ethernet40(S)
0 PortChannel0	LACP(A)(Up)	Ethernet0(S)	Ethernet4(S)
8 PortChannel8	LACP(A)(Up)	Ethernet8(S)	Ethernet12(S)



show interface status

This command displays some more fields such as Lanes, Speed, MTU, Type, Asymmetric PFC status and also the operational and administrative status of the interfaces

- Usage:

```
show interfaces status [<interface_name>]
```

- Example (show interface status of all interfaces):

```
admin@sonic:~$ show interfaces status
```

Interface	Lanes	Speed	MTU	Alias	Oper	Admin	Type
	Asym PFC						
Ethernet0	49,50,51,52	off	100G	9100	hundredGigE1/1	down	up
N/A	53,54,55,56	off					
Ethernet4	57,58,59,60		100G	9100	hundredGigE1/2	down	up
N/A	off						
Ethernet8			100G	9100	hundredGigE1/3	down	down
N/A							

<continues to display all the interfaces>

- Example (to only display the status for interface Ethernet0):

```
admin@sonic:~$ show interface status Ethernet0
```

Interface	Lanes	Speed	MTU	Alias	Oper	Admin
Ethernet0	101,102	40G	9100	fortyGigE1/1/1	up	up

- Example (to only display the status for range of interfaces):

```
admin@sonic:~$ show interfaces status Ethernet8,Ethernet168-180
```

Interface	Lanes	Speed	MTU	Alias	Oper	Admin
Type	Asym PFC					
Ethernet8	49,50,51,52		100G	9100	hundredGigE3	down
N/A	N/A					down
Ethernet168	9,10,11,12		100G	9100	hundredGigE43	down
N/A	N/A					down
Ethernet172	13,14,15,16		100G	9100	hundredGigE44	down
N/A	N/A					down
Ethernet176	109,110,111,112		100G	9100	hundredGigE45	down
N/A	N/A					down
Ethernet180	105,106,107,108		100G	9100	hundredGigE46	down
N/A	N/A					down



show interface switchport status

This command displays switchport modes status of the interfaces

- Usage:

```
show interfaces switchport status
```

- Example (show interface switchport status of all interfaces):

```
admin@sonic:~$ show interfaces switchport status
```

Interface	Mode
-----	-----
Ethernet0	access
Ethernet4	trunk
Ethernet8	routed

<continues to display all the interfaces>

show interface switchport config

This command displays switchport modes configuration of the interfaces

- Usage:

```
show interfaces switchport config
```

- Example (show interface switchport config of all interfaces):

```
admin@sonic:~$ show interfaces switchport config
```

Interface	Mode	Untagged -	Tagged --
-----	-----	-----2	-----
Ethernet0	access	3	
Ethernet4	trunk		4,5,6
Ethernet8	routed		

<continues to display all the interfaces>

For details please refer Switchport Mode HLD to know more about this command.



show interfaces transceiver

This command is already explained here

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.22.2 Interface Config Commands

This sub-section explains the following list of configuration on the interfaces.

1. ip - To add or remove IP address for the interface
2. pfc - to set the PFC configuration for the interface
3. shutdown - to administratively shut down the interface
4. speed - to set the interface speed
5. startup - to bring up the administratively shutdown interface
6. breakout - to set interface breakout mode
7. autoneg - to set interface auto negotiation mode
8. advertised-speeds - to set interface advertised speeds
9. advertised-types - to set interface advertised types
10. type - to set interface type
11. mpls - To add or remove MPLS operation for the interface
12. loopback-action - to set action for packet that ingress and gets routed on the same IP interface
13. link-training - to set interface link-training mode

From 201904 release onwards, the “config interface” command syntax is changed and the format is as follows:

- config interface interface_subcommand <interface_name> i.e Interface name comes after the subcommand
- Ex: config interface startup Ethernet63

The syntax for all such interface_subcommands are given below under each command

NOTE: In older versions of SONiC until 201811 release, the command syntax was
config interface <interface_name> interface_subcommand

config interface ip add <interface_name> <ip_addr> [default_gw] (Versions >= 201904)

config interface <interface_name> ip add <ip_addr> (Versions <= 201811)

This command is used for adding the IP address for an interface. IP address for either physical interface or for portchannel or for VLAN interface or for Loopback interface can be configured using this



command. While configuring the IP address for the management interface “eth0”, users can provide the default gateway IP address as an optional parameter from release 201911.

- Usage:

Versions >= 201904

```
config interface ip add <interface_name> <ip_addr>
```

Versions <= 201811

```
config interface <interface_name> ip add <ip_addr>
```

- Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface ip add Ethernet63 10.11.12.13/24 admin@sonic:~$ sudo config interface ip add eth0 20.11.12.13/24 20.11.12.254
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface Ethernet63 ip add 10.11.12.13/24
```

VLAN interface names take the form of `vlan<vlan_id>` . E.g., VLAN 100 will be named `vlan100` •

Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface ip add Vlan100 10.11.12.13/24
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface vlan100 ip add 10.11.12.13/24
```

config interface ip remove <interface_name> <ip_addr> (Versions >= 201904)

config interface <interface_name> ip remove <ip_addr> (Versions <= 201811) •

Usage:

Versions >= 201904

```
config interface ip remove <interface_name> <ip_addr>
```

Versions <= 201811

```
config interface ip remove <interface_name> <ip_addr>
```

- Example:

Versions >= 201904



```
admin@sonic:~$ sudo config interface ip remove Ethernet63 10.11.12.13/24 admin@sonic:~$ sudo
config interface ip remove eth0 20.11.12.13/24
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface Ethernet63 ip remove 10.11.12.13/24
```

VLAN interface names take the form of `vlan<vlan_id>` . E.g., VLAN 100 will be named `vlan100` •

Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface ip remove vlan100 10.11.12.13/24
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface vlan100 ip remove 10.11.12.13/24
```

config interface pfc priority <interface_name> (on | off)

This command is used to set PFC on a given priority of a given interface to either “on” or “off”. Once it is successfully configured, it will show current losses priorities on the given interface. Otherwise, it will show error information

- Example: *Versions >= 201904*

```
admin@sonic:~$ sudo config interface pfc priority Ethernet0 3 off
```

<u>Interface</u>	<u>Lossless priorities</u>
Ethernet0	4

```
admin@sonic:~$ sudo config interface pfc priority Ethernet0 8 off Usage: pfc config
priority [OPTIONS] STATUS INTERFACE PRIORITY
```

Error: Invalid value for "priority": invalid choice: 8. (choose from 0, 1, 2, 3, 4, 5, 6, 7)

```
admin@sonic:~$ sudo config interface pfc priority Ethernet101 3 off Cannot find interface Ethernet101
```

```
admin@sonic:~$ sudo config interface pfc priority Ethernet0 3 on
```

<u>Interface</u>	<u>Lossless priorities</u>
-----	-----
Ethernet0	3,4

config interface pfc asymmetric <interface_name> (Versions >= 201904)

config interface <interface_name> pfc asymmetric (Versions <= 201811)

This command is used for setting the asymmetric PFC for an interface to either “on” or “off”. Once it is configured, use “show interfaces status” to check the same.



- Usage:

Versions >= 201904

```
config interface pfc asymmetric <interface_name> on/off (for 201904+ version)
```

Versions <= 201811

```
config interface <interface_name> pfc asymmetric on/off (for 201811- version)
```

- Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface pfc asymmetric Ethernet60 on
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface Ethernet60 pfc asymmetric on
```

config interface shutdown <interface_name> (Versions >= 201904)

config interface <interface_name> shutdown (Versions <= 201811)

This command is used to administratively shut down either the Physical interface or port channel interface. Once if it is configured, use “show interfaces status” to check the same.

- Usage:

Versions >= 201904

```
config interface shutdown <interface_name> (for 201904+ version)
```

Versions <= 201811

```
config interface <interface_name> shutdown (for 201811- version)
```

- Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface shutdown Ethernet63
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface Ethernet63 shutdown
```

shutdown multiple interfaces

```
admin@sonic:~$ sudo config interface shutdown Ethernet8,Ethernet16-20,Ethernet32
```

config interface startup <interface_name> (Versions >= 201904)

config interface <interface_name> startup (Versions <= 201811)



This command is used for administratively bringing up the Physical interface or port channel interface. Once if it is configured, use “show interfaces status” to check the same.

- Usage:

Versions >= 201904

```
config interface startup <interface_name> (for 201904+ version)
```

Versions <= 201811

```
config interface <interface_name> startup (for 201811- version)
```

- Example:

Versions >= 201904

```
admin@sonic:~$ sudo config interface startup Ethernet63
```

Versions <= 201811

```
admin@sonic:~$ sudo config interface Ethernet63 startup
```

startup multiple interfaces

```
admin@sonic:~$ sudo config interface startup Ethernet8,Ethernet16-20,Ethernet32
```

config interface <interface_name> speed (Versions >= 202006)

Dynamic breakout feature is supported in SONiC from 202006 version. User can configure any speed specified under “breakout_modes” keys for the parent interface in the platform-specific port configuration file (i.e. platform.json).

For example for a breakout mode of 2x50G[25G,10G] the default speed is 50G but the interface also supports 25G and 10G.

Refer DPB HLD DOC to know more about this command.

config interface speed <interface_name> (Versions >= 201904)

config interface <interface_name> speed (Versions <= 201811)

This command is used to configure the speed for the Physical interface. Use the value 40000 for setting it to 40G and 100000 for 100G. Users need to know the device to configure it properly.

- Usage:

Versions >= 201904

```
config interface speed <interface_name> <speed_value>
```

Versions <= 201811

```
config interface <interface_name> speed <speed_value>
```

- Example (Versions >= 201904):



```
admin@sonic:~$ sudo config interface speed Ethernet63 40000
```

- Example (Versions <= 201811):

```
admin@sonic:~$ sudo config interface Ethernet63 speed 40000
```

config interface transceiver lpmode

This command is used to enable or disable low-power mode for an SFP transceiver

- Usage:

```
config interface transceiver lpmode <interface_name> (enable | disable)
```

- Examples:

```
user@sonic~$ sudo config interface transceiver lpmode Ethernet0 enable Enabling low-power mode for port Ethernet0... OK
```

```
user@sonic~$ sudo config interface transceiver lpmode Ethernet0 disable Disabling low-power mode for port Ethernet0... OK
```

config interface transceiver frequency

This command is used to set the coherent module laser frequency

- Usage:

```
config interface transceiver frequency <interface_name> <frequency_value>
```

- Examples:

```
user@sonic:~# config interface transceiver frequency Ethernet0 196100 Setting laser frequency to 196100 GHz on port Ethernet0
```

config interface transceiver tx_power

This command is used to set the coherent module tx power

- Usage:

```
config interface transceiver tx_power <interface_name> <tx_power_value>
```

- Examples:

```
user@sonic:~# config interface transceiver tx_power Ethernet0 -- -10.0 Setting target Tx output power to -10.0 dBm on port Ethernet0
```

config interface transceiver reset

This command is used to reset an SFP transceiver

- Usage:

```
config interface transceiver reset <interface_name>
```



- Examples:

```
user@sonic~$ sudo config interface transceiver reset Ethernet0 Resetting port
Ethernet0... OK
```

config interface transceiver dom

This command is used to configure the Digital Optical Monitoring (DOM) for an interface.

- Usage:

```
config interface transceiver dom <interface_name> (enable | disable)
```

- Examples:

```
user@sonic~$ sudo config interface transceiver dom Ethernet0 enable
```

```
user@sonic~$ sudo config interface transceiver dom Ethernet0 disable
```

config interface mtu <interface_name> (Versions >= 201904)

This command is used to configure the mtu for the Physical interface. Use the value 1500 for setting max transfer unit size to 1500 bytes.

- Usage:

Versions >= 201904

```
config interface mtu <interface_name> <mtu_value>
```

- Example (Versions >= 201904):

```
admin@sonic:~$ sudo config interface mtu Ethernet64 1500
```

config interface tpid <interface_name> (Versions >= 202106)

This command is used to configure the TPID for the Physical/PortChannel interface. default is 0x8100. Other allowed values if supported by HW SKU (0x9100, 0x9200, 0x88A8).

- Usage:

Versions >= 202106

```
config interface tpid <interface_name> <tpid_value>
```

- Example (Versions >= 202106):

```
admin@sonic:~$ sudo config interface tpid Ethernet64 0x9200
```

config interface breakout (Versions >= 202006)

This command is used to set active breakout mode available for user-specified interface based on the platform-specific port configuration file(i.e. platform.json) and the current mode set for the interface.

Based on the platform.json and the current mode set in interface, this command acts on setting breakout mode for the interface.



Double tab i.e. to see the available breakout option customized for each interface provided by the user.

- Usage:

```
sudo config interface breakout --help
Usage: config interface breakout [OPTIONS] <interface_name> MODE

Set interface breakout mode

Options:
  -f, --force-remove-dependencies      Clear all depenedecies internally first. load predefined user
                                        configuration (alias, lanes, speed etc) first.
  -l, --load-predefined-config
  -y, --yes                            Enable verbose output
  -v, --verbose -?, -h,                Show this message and exit.
  --help
```

- Example :

```
admin@sonic:~$ sudo config interface breakout Ethernet0 <tab><tab> <tab provides
option for breakout mode>
1x100G[40G] 2x50G          4x25G[10G]
```

This command also provides “-force-remove-dependencies/-f” option to CLI, which will automatically determine and remove the configuration dependencies using Yang models.

```
admin@sonic:~$ sudo config interface breakout Ethernet0 4x25G[10G] -f -l -v -y
```

For details please refer DPB HLD DOC to know more about this command.

config interface autoneg <interface_name> (Versions >= 202106)

This command is used to set port auto negotiation mode.

- Usage:

```
sudo config interface autoneg --help
Usage: config interface autoneg [OPTIONS] <interface_name> <mode>

Set interface auto negotiation mode

Options:
  -v, --verbose -h, -?, Enable verbose output
  --help                  Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface autoneg Ethernet0 enabled

admin@sonic:~$ sudo config interface autoneg Ethernet0 disabled
```




config interface advertised-speeds <interface_name> (Versions >= 202106)

This command is used to set port advertised speed.

- Usage:

```
sudo config interface advertised-speeds --help
Usage: config interface advertised-speeds [OPTIONS] <interface_name> <speed_list>

Set interface advertised speeds

Options:
  -v, --verbose  -h, -?, Enable verbose output
  --help          Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface advertised-speeds Ethernet0 all

admin@sonic:~$ sudo config interface advertised-speeds Ethernet0 50000,100000
```

config interface advertised-types <interface_name> (Versions >= 202106)

This command is used to set port advertised interface types.

- Usage:

```
sudo config interface advertised-types --help
Usage: config interface advertised-types [OPTIONS] <interface_name> <interface_type_list>

Set interface advertised types

Options:
  -v, --verbose  -h, -?, Enable verbose output
  --help          Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface advertised-types Ethernet0 all

admin@sonic:~$ sudo config interface advertised-types Ethernet0 CR,CR4
```

config interface type <interface_name> (Versions >= 202106)

This command is used to set port interface type.

- Usage:

```
sudo config interface type --help
Usage: config interface type [OPTIONS] <interface_name> <interface_type_value>

Set interface type

Options:
  -v, --verbose  -h, -?, Enable verbose output
  --help          Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface type Ethernet0 CR4
```



config interface cable_length (Versions >= 202006)

This command is used to configure the length of the cable connected to a port. The cable_length is in unit of meters and must be suffixed with “m”.

For details please refer dynamic buffer management

config interface lossless_pg (Versions >= 202006)

This command is used to configure the priority groups on which lossless traffic runs.

For details please refer dynamic buffer management

config interface headroom_override (Versions >= 202006)

This command is used to configure a static buffer profile on a port's lossless priorities. There shouldn't be any lossless_pg configured on the port when configuring headroom_override . The port's headroom won't be updated after headroom_override has been configured on the port.

For details please refer dynamic buffer management

config interface mpls add <interface_name> (Versions >= 202106)

This command is used for adding MPLS operation on the interface. MPLS operation for either physical, portchannel, or VLAN interface can be configured using this command.

- Usage:

```
sudo config interface mpls add --help
Usage: config interface mpls add [OPTIONS] <interface_name>

Add MPLS operation on the interface

Options:
  -?, -h, --help  Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface mpls add Ethernet4
```

config interface mpls remove <interface_name> (Versions >= 202106)

This command is used for removing MPLS operation on the interface. MPLS operation for either physical, portchannel, or VLAN interface can be configured using this command.

- Usage:

```
sudo config interface mpls remove --help
Usage: config interface mpls remove [OPTIONS] <interface_name>

Remove MPLS operation from the interface

Options:
  -?, -h, --help  Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface mpls remove Ethernet4
```



config interface ip loopback-action <interface_name> (Versions >= 202205)

This command is used for setting the action being taken on packets that ingress and get routed on the same IP interface. Loopback action can be set on IP interface from type physical, portchannel, VLAN interface and VLAN subinterface. Loopback action can be drop or forward.

- Usage:

```
config interface ip loopback-action --help
Usage: config interface ip loopback-action [OPTIONS] <interface_name> <action>

Set IP interface loopback action

Options:
  -?, -h, --help  Show this message and exit.
```

- Example:

```
admin@sonic:~$ config interface ip loopback-action Ethernet0 drop admin@sonic:~$ config
interface ip loopback-action Ethernet0 forward
```

config interface link-training <interface_name> (Versions >= 202211)

This command is used for setting link-training mode of a interface.

- Usage:

```
sudo config interface link-training --help
Usage: config interface link-training [OPTIONS] <interface_name> <mode>

Set interface link-training mode

Options:
  -v, --verbose  -h, -?, Enable verbose output
  --help        Show this message and exit.
```

- Example:

```
admin@sonic:~$ sudo config interface link-training Ethernet0 on admin@sonic:~$
sudo config interface link-training Ethernet0 off
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.23 Interface Naming Mode

2.1.23.1 Interface naming mode show commands

This command displays the current interface naming mode. Interface naming mode originally set to 'default'. Interfaces are referenced by default SONiC interface names. Users can change the naming_mode using “config interface_naming_mode” command.

show interfaces naming_mode

This command displays the current interface naming mode



- Usage:

```
show interfaces naming_mode
```

- Examples:

```
admin@sonic:~$ show interfaces naming_mode default
```

- “default” naming mode will display all SONiC interface names in ‘show’ commands and accept SONiC interface names as parameters in ‘config’ commands

```
admin@sonic:~$ show interfaces naming_mode alias
```

- “alias” naming mode will display all hardware vendor interface aliases in ‘show’ commands and accept hardware vendor interface aliases as parameters in ‘config’ commands

2.1.23.2 Interface naming mode config commands

config interface_naming_mode

This command is used to change the interface naming mode. Users can select between default mode (SONiC interface names) or alias mode (Hardware vendor names). The user must log out and log back in for changes to take effect. Note that the newly-applied interface mode will affect all interface-related show/config commands.

NOTE: Some platforms do not support alias mapping. In such cases, this command is not applicable. Such platforms always use the same SONiC interface names.

- Usage:

```
config interface_naming_mode (default | alias)
```

- Interface naming mode is originally set to ‘default’. Interfaces are referenced by default SONiC interface names:

- Example:

```
admin@sonic:~$ show interfaces naming_mode default
```

```
admin@sonic:~$ show interface status Ethernet0
```

Interface ----	Lanes -	Speed	MTU	Alias	Oper -	Admin
-----	-----	-----	-----	-----	-----	-----
Ethernet0	101,102	40G	9100	fortyGigE1/1/1	up	up

```
admin@sonic:~$ sudo config interface_naming_mode alias Please logout and log back in for changes take effect.
```

- After user logs out and logs back in again, interfaces will then referenced by hardware vendor aliases:

```
admin@sonic:~$ show interfaces naming_mode alias
```



```
admin@sonic:~$ sudo config interface fortyGigE1/1/1 shutdown admin@sonic:~$
show interface status fortyGigE1/1/1
```

Interface ----	Lanes -	Speed	MTU	Alias	Oper -	Admin
-----	-----	-----	-----	-----	-----	-----
Ethernet0	101,102	40G	9100	fortyGigE1/1/1	down	down

[Go Back To Beginning of the document or Beginning of this section](#)

2.1.24 Interface Vrf binding

2.1.24.1 Interface vrf bind & unbind config commands

config interface vrf bind

This command is used to bind a interface to a vrf. By default, all L3 interfaces will be in default vrf. Above vrf bind command will let users bind interface to a vrf.

- Usage:

```
config interface vrf bind <interface_name> <vrf_name>
```

config interface vrf unbind

This command is used to ubind a interface from a vrf. This will move the interface to default vrf.

- Usage:

```
config interface vrf unbind <interface_name>
```

2.1.24.2 Interface vrf binding show commands

To display interface vrf binding information, user can use show vrf command. Please refer sub-section Vrf-show-command.

[Go Back To Beginning of the document or Beginning of this section](#)

2.1.25 IP / IPv6

2.1.25.1 IP show commands

This sub-section explains the various IP protocol specific show commands that are used to display the following.

1. routes
2. bgp details - Explained in the bgp section
3. IP interfaces
4. prefix-list
5. protocol

2.1.25.1.1 show ip route

This command displays either all the route entries from the routing table or a specific route.

- Usage:

```
show ip route [<vrf-name>] [<ip_address>]
```

- Example:



```
admin@sonic:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B -
      BGP, P - PIM, A - Babel,
      > - selected route, * - FIB route
S>* 0.0.0.0/0 [200/0] via 10.11.162.254, eth0 C>* 1.1.0.0/16 is
      directly connected, Vlan100
C>* 10.1.1.0/31 is directly connected, Ethernet112 C>* 10.1.1.2/31 is
      directly connected, Ethernet116 C>* 10.11.162.0/24 is directly
      connected, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 240.127.1.0/24 is directly connected, docker0
```

- Optionally, you can specify an IP address in order to display only routes to that particular IP address
- Example:

```
admin@sonic:~$ show ip route 10.1.1.0 Routing entry for 10.1.1.0/31
      Known via "connected", distance 0, metric 0, best * directly connected, Ethernet112
```

- Vrf-name can also be specified to get IPv4 routes programmed in the vrf.

- Example:

```
admin@sonic:~$ show ip route vrf Vrf-red
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
      T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR, f - OpenFabric,
      > - selected route, * - FIB route VRF Vrf-red:
C>* 11.1.1.1/32 is directly connected, Loopback11, 21:50:47 C>* 100.1.1.0/24 is directly connected,
      Vlan100, 03w1d06h

admin@sonic:~$ show ip route vrf Vrf-red 11.1.1.1/32 Routing entry for 11.1.1.1/32
      Known via "connected", distance 0, metric 0, vrf Vrf-red, best Last update 21:57:53 ago
      * directly connected, Loopback11
```

2.1.25.1.2 show ip interfaces

This command displays the details about all the Layer3 IP interfaces in the device for which IP address has been assigned. The type of interfaces include the following.

1. Front panel physical ports.
2. PortChannel.
3. VLAN interface.
4. Loopback interfaces
5. docker interface and
6. management interface



- Usage:

```
show ip interfaces
```

- Example:

```
admin@sonic:~$ show ip interfaces
```

Interface	Master	IPv4 address/mask	Admin/Oper	BGP
Neighbor	Neighbor IP	Flags		
-----	-----	-----	-----	
-----	-----	1.0.0.1/32		
Loopback0			up/up	N/A
N/A Loopback11		11.1.1.1/32		
N/A Loopback100	Vrf-red		up/up	N/A
N/A		100.0.0.1/32		
PortChannel01	Vrf-blue		up/up	N/A
10.0.0.57		10.0.0.56/31		
PortChannel02			up/down	DEVICE1
10.0.0.59		10.0.0.58/31		
PortChannel03			up/down	DEVICE2
10.0.0.61		10.0.0.60/31		
PortChannel04			up/down	DEVICE3
10.0.0.63 Vlan100		10.0.0.62/31		
N/A			up/down	DEVICE4
Vlan1000	Vrf-red	1001.1.1/24		
N/A			up/up	N/A
docker0		192.168.0.1/27		
N/A			up/up	N/A
eth0		240.127.1.1/24		
N/A lo			up/down	N/A
N/A		10.3.147.252/23		
			up/up	N/A
		127.0.0.1/8		
			up/up	N/A



2.1.25.1.3 show ip protocol

This command displays the route-map that is configured for the routing protocol. Refer the routing stack Quagga Command Reference or FRR Command Reference to know more about this command.

- Usage:

```
show ip protocol
```

- Example:

```
admin@sonic:~$ show ip protocol
Protocol      : route-map
system       : none
kernel       : none
connected    : none
static       : none
rip          : none
ripng        : none
ospf         : none
ospf6        : none
isis         : none
bgp          : RM_SET_SRC
pim          : none
hsls         : none
olsr         : none
babel        : none
any          : none
```

2.1.25.2 IPv6 show commands

This sub-section explains the various IPv6 protocol specific show commands that are used to display the following.

1. routes
2. IPv6 bgp details - Explained in the bgp section
3. IP interfaces
4. protocol

show ipv6 route

This command displays either all the IPv6 route entries from the routing table or a specific IPv6 route.

- Usage:

```
show ipv6 route [<vrf-name>] [<ipv6_address>]
```

- Example:

```
admin@sonic:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
      > - selected route, * - FIB route
C>* ::1/128 is directly connected, lo
C>* 2018:2001::/126 is directly connected, Ethernet112 C>* 2018:2002::/126 is directly connected, Ethernet116 C>*
fc00:1::32/128 is directly connected, lo
C>* fc00:1::102/128 is directly connected, lo C>* fc00:2::102/128 is directly connected, eth0 C *
fe80::/64 is directly connected, Vlan100
C * fe80::/64 is directly connected, Ethernet112 C * fe80::/64 is directly connected, Ethernet116 C * fe80::/64 is directly
connected, Bridge
C * fe80::/64 is directly connected, PortChannel0011 C>* fe80::/64 is directly connected, eth0
```

- Optionally, you can specify an IPv6 address in order to display only routes to that particular IPv6 address
- Example:

```
admin@sonic:~$ show ipv6 route fc00:1::32 Routing entry for fc00:1::32/128
```




Known via "connected", distance 0, metric 0, best * directly connected, lo

Vrf-name can also be specified to get IPv6 routes programmed in the vrf.

- Example:

```
admin@sonic:~$ show ipv6 route vrf Vrf-red
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR, f -
OpenFabric,
> - selected route, * - FIB route VRF Vrf-red:
C>* 1100::1/128 is directly connected, Loopback11, 21:50:47 C>* 100::1/128 is directly connected,
Vlan100, 03w1d06h
C>* fe80::/64 is directly connected, Loopback11, 21:50:47 C>* fe80::/64 is directly connected,
Vlan100, 03w1d06h

admin@sonic:~$ show ipv6 route vrf Vrf-red 1100::1/128 Routing entry for 1100::1/128
Known via "connected", distance 0, metric 0, vrf Vrf-red, best Last update 21:57:53 ago
* directly connected, Loopback11
```

show ipv6 interfaces

This command displays the details about all the Layer3 IPv6 interfaces in the device for which IPv6 address has been assigned. The type of interfaces include the following.

1. Front panel physical ports.
2. PortChannel.
3. VLAN interface.
4. Loopback interfaces
5. management interface

- Usage:

```
show ipv6 interfaces
```

- Example:

```
admin@sonic:~$ show ipv6 interfaces
Interface      Master      IPv6 address/mask      Admin/
Oper   BGP Neighbor   Neighbor IP
-----
Bridge
N/A          N/A          fe80::7c45:1dff:fe08:cdd%Bridge/64      up/up
Loopback11    Vrf-red     1100::1/128             up/up
PortChannel01          fc00::71/126            up/down
DEVICE1       fc00::72          fc00::75/126            up/down
PortChannel02          fc00::76                up/down
DEVICE2       fc00::76          fc00::79/126            up/down
PortChannel03          fc00::7a                up/down
DEVICE3       fc00::7a          fc00::7d/126            up/down
PortChannel04          fc00::7e                up/down
DEVICE4       fc00::7e          100::1/112              up/up
Vlan100       Vrf-red     100::1/112              up/up
N/A          N/A          fe80::eef4:bbff:fe08:880a%Vlan100/64
eth0
N/A          N/A          fe80::eef4:bbff:fe08:880a%eth0/64      up/up
lo
N/A          N/A          fc00::1::32/128         up/up
```





show ipv6 protocol

This command displays the route-map that is configured for the IPv6 routing protocol. Refer the routing stack Quagga Command Reference or FRR Command Reference to know more about this command.

- Usage:

```
show ipv6 protocol
```

- Example:

```
admin@sonic:~$ show ipv6 protocol
Protocol      : route-map
-----
system       : none
kernel       : none
connected    : none
static rip    : none
ripng        : none
ospf         : none
ospf6        : none
isis bgp     : none
pim          : none
hsls        : RM_SET_SRC6
olsr         : none
babel        : none
any          : none
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.26 IPv6 Link Local

2.1.26.1 IPv6 Link Local config commands

This section explains all the commands that are supported in SONiC to configure IPv6 Link-local.

config interface ipv6 enable use-link-local-only <interface_name>

This command enables user to enable an interface to forward L3 traffic with out configuring an address. This command creates the routing interface based on the auto generated IPv6 link-local address. This command can be used even if an address is configured on the interface.

- Usage:

```
config interface ipv6 enable use-link-local-only <interface_name>
```

- Example:

```
admin@sonic:~$ sudo config interface ipv6 enable use-link-local-only Vlan206 admin@sonic:~$ sudo
config interface ipv6 enable use-link-local-only PortChannel007
admin@sonic:~$ sudo config interface ipv6 enable use-link-local-only Ethernet52
```



config interface ipv6 disable use-link-local-only <interface_name>

This command enables user to disable use-link-local-only configuration on an interface.

- Usage:

```
config interface ipv6 disable use-link-local-only <interface_name>
```

- Example:

```
admin@sonic:~$ sudo config interface ipv6 disable use-link-local-only Vlan206 admin@sonic:~$ sudo
config interface ipv6 disable use-link-local-only PortChannel007
admin@sonic:~$ sudo config interface ipv6 disable use-link-local-only Ethernet52
```

config ipv6 enable link-local

This command enables user to enable use-link-local-only command on all the interfaces globally.

- Usage:

```
sudo config ipv6 enable link-local
```

- Example:

```
admin@sonic:~$ sudo config ipv6 enable link-local
```

config ipv6 disable link-local

This command enables user to disable use-link-local-only command on all the interfaces globally.

- Usage:

```
sudo config ipv6 disable link-local
```

- Example:

```
admin@sonic:~$ sudo config ipv6 disable link-local
```

2.1.26.2 IPv6 Link Local show commands

show ipv6 link-local-mode

This command displays the link local mode of all the interfaces.

- Usage:

```
show ipv6 link-local-mode
```

- Example:

```
root@sonic:/home/admin# show ipv6 link-local-mode
+-----+-----+
| Interface Name | Mode      |
+=====+=====+
| Ethernet16     | Disabled  |
+-----+-----+
| Ethernet18     | Enabled   |
+-----+-----+
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.27 Kubernetes

2.1.27.1 Kubernetes show commands

show kubernetes server config

This command displays the kubernetes server configuration, if any, else would report as not configured.

- Usage:

```
show kubernetes server config
```

- Example:

```
admin@sonic:~$ show kubernetes server config
ip            port      insecure   disable
-----
10.3.157.24    6443      True       False
```

show kubernetes server status

This command displays the kubernetes server status.

- Usage:

```
show kubernetes server status
```

- Example:

```
admin@sonic:~$ show kubernetes server status
ip            port      connected  update-time
-----
10.3.157.24    6443      true       2020-11-15 18:25:05
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.28 LDAP

2.1.28.1 show LDAP global commands

This command displays the global LDAP configuration that includes the following parameters: base_dn, bind_password, bind_timeout, version, port, timeout.

- Usage:

```
show ldap global
```

- Example:

```
admin@sonic:~$ show ldap global
base-dn      Ldap user base dn <string> LDAP
bind-dn      global bind dn <string>
bind-password Shared secret used for encrypting the communication
<password>
bind-timeout  Ldap bind timeout <0-120>
port         TCP port to communicate with LDAP server <1-65535> Ldap timeout
timeout      duration in sec <1-60>
version      Ldap version <1-3>
```



2.1.28.2 LDAP global config commands

These commands are used to configure the LDAP global parameters

- Usage:

```
config ldap global
```

- Example:

```
admin@sonic:~$ config ldap global

host base- <ADDRESS> --prio <1 - 8>
dn          Ldap user base dn <string>
bind-dn     LDAP global bind dn <string>

bind-password Shared secret used for encrypting the communication <password> Ldap bind timeout
bind-timeout port <0-120>
timeout      TCP port to communicate with LDAP server <1-65535> Ldap timeout
version      duration in sec <1-60>
              Ldap version <1-3>
```

2.1.28.3 show LDAP server commands

This command displays the global LDAP configuration that includes the following parameters: base_dn, bind_password, bind_timeout, version, port, timeout.

- Usage:

```
show ldap-server
```

- Example:

```
admin@sonic:~$ show ldap-server

hostname      Ldap hostname or IP of the configured LDAP server
priority      priority for the relevant LDAP server <1-8>
```



2.1.28.4 LDAP server config commands

These commands are used to manage the LDAP servers in the system, they are created in correspondence to the global config parameters mentioned earlier.

- Usage:

```
config ldap-server
```

- Example:

```
admin@sonic:~$ config ldap-server
```

```
add          Add a new LDAP server <ip> --priority <1-8>
```

```
delete       Delete an existing LDAP server from the list <ip> --priority <1-8>
```

```
update       Update an existing LDAP server
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.1.29 Linux Kernel Dump

This section demonstrates the show commands and configuration commands of Linux kernel dump mechanism in SONiC.

2.1.29.1 Linux Kernel Dump show commands

show kdump config

This command shows the configuration of Linux kernel dump.

- Usage:

```
show kdump config
```

- Example:

```
admin@sonic:$ show kdump config Kdump
```

```
administrative mode: Disabled Kdump
```

```
operational mode: Unready
```

```
Kdump memory reservation: 0M-2G:256M,2G-4G:320M,4G-8G:384M,8G-:448M Maximum number of Kdump files: 3
```

show kdump files

This command shows the Linux kernel core dump files and dmesg files which are generated by kernel dump tool.

- Usage:

```
show kdump files
```

- Example:

```
admin@sonic:~$ show kdump files
```

```
Kernel core dump files
```

```
Kernel dmesg files
```

```
/var/crash/202106242344/kdump.2021062
```

```
42344 /var/crash/202106242344/
```

```
dmesg.202106242344
```

```
/var/crash/202106242337/kdump.2021062423
```

```
37 /var/crash/202106242337/
```

```
dmesg.202106242337
```



show kdump logging <file_name> <num_of_lines>

By default, this command will show the last 10 lines of latest dmesg file. This command can also accept a specific file name and number of lines as arguments.

- Usage:

```
show kdump logging
```

- Example:

```
admin@sonic:~$ show kdump logging
[ 157.642053] RSP: 002b:00007fff1beee708 EFLAGS: 00000246 ORIG_RAX:
0000000000000001
[ 157.732635] RAX: ffffffffda RBX: 0000000000000002 RCX: 00007fc3887d4504 [ 157.818015] RDX:
0000000000000002 RSI: 000055d388eceb40 RDI: 0000000000000001 [ 157.903401] RBP: 000055d388eceb40
R08: 000000000000000a R09: 00007fc3888255f0 [ 157.988784] R10: 000000000000000a R11:
0000000000000246 R12: 00007fc3888a6760 [ 158.074166] R13: 0000000000000002 R14: 00007fc3888a1760
R15: 0000000000000002 [ 158.159553] Modules linked in: nft_chain_route_ipv6(E) nft_chain_route_ipv4(E)
xt_TCPMSS(E) dummy(E) team_mode_loadbalance(E) team(E) sx_bfd(OE) sx_netdev(OE) psample(E)
sx_core(OE) 8021q(E) garp(E) mrp(E) mst_pciconf(OE) mst_pci(OE) xt_hl(E) xt_tcpudp(E) ip6_tables(E)
nft_compat(E) nft_chain_nat_ipv4(E) nf_nat_ipv4(E) nft_counter(E) xt_conntrack(E) nf_nat(E) jc42(E)
nf_conntrack_netlink(E) nf_conntrack(E) nf_defrag_ipv6(E) nf_defrag_ipv4(E) libcrc32c(E) xfrm_user(E)
xfrm_algo(E) mlxsw_minimal(E) mlxsw_i2c(E) i2c_mux_reg(E) i2c_mux(E) i2c_mlxcp(E) leds_mlxreg(E)
mlxreg_io(E) mlxreg_hotplug(E) mei_wdt(E) evdev(E) intel_rapl(E) x86_pkg_temp_thermal(E)
intel_powerclamp(E) kvm_intel(E) mlx_platform(E) kvm(E) irqbypass(E) crct10dif_pclmul(E) crc32_pclmul(E)
ghash_clmulni_intel(E) intel_cstate(E) intel_uncore(E)
[ 159.016731] intel_rapl_perf(E) pcspkr(E) sg(E) iTCO_wdt(E) iTCO_vendor_support(E) mei_me(E) mei(E)
bonding(E) pcc_cpufreq(E) video(E) button(E) ebt_vlan(E) ebtable_broute(E) bridge(E) stp(E) llc(E)
ebtable_nat(E) ebtable_filter(E) ebtables(E) nf_tables(E) nfnetlink(E) xdpe12284(E) at24(E) ledtrig_timer(E)
tmp102(E) lm75(E) drm(E) coretemp(E) max1363(E) industrialio_triggered_buffer(E) kfifo_buf(E)
industrialio(E) tps53679(E) fuse(E) pmbus(E) pmbus_core(E) i2c_dev(E) configfs(E) ip_tables(E) x_tables(E)
autofs4(E) loop(E) ext4(E) crc16(E) mbcache(E) jbd2(E) crc32c_generic(E) fscrypto(E) ecb(E) crypto_simd(E)
cryptd(E) glue_helper(E) aes_x86_64(E) nvme(E) nvme_core(E) nls_utf8(E) nls_cp437(E) nls_ascii(E) vfat(E)
fat(E) overlay(E) squashfs(E) zstd_decompress(E) xxhash(E) sd_mod(E) gpio_ich(E) ahci(E)
[ 159.864532] libahci(E) mlxsw_core(E) devlink(E) ehci_pci(E) ehci_hcd(E) crc32c_intel(E) libata(E)
i2c_i801(E) scsi_mod(E) usbcore(E) usb_common(E)
lpc_ich(E) mfd_core(E) e1000e(E) fan(E) thermal(E) [ 160.075846]
CR2: 0000000000000000
```




You can specify a file name in order to show its last 10 lines.

- Example:

```
admin@sonic:~$ show kdump logging dmesg.202106242337
[ 654.120195] RSP: 002b:00007ffe697690f8 EFLAGS: 00000246 ORIG_RAX:
0000000000000001
[ 654.210778] RAX: ffffffffda RBX: 0000000000000002 RCX: 00007fcfa27b504 [ 654.296157] RDX:
0000000000000002 RSI: 000055a6e4d1b3f0 RDI: 0000000000000001 [ 654.381543] RBP: 000055a6e4d1b3f0
R08: 000000000000000a R09: 00007fcfa2cc5f0 [ 654.466925] R10: 000000000000000a R11:
0000000000000246 R12: 00007fcfa34d760 [ 654.552310] R13: 0000000000000002 R14: 00007fcfa348760
R15: 0000000000000002 [ 654.637694] Modules linked in: binfmt_misc(E) nft_chain_route_ipv6(E)
nft_chain_route_ipv4(E) xt_TCPMSS(E) dummy(E) team_mode_loadbalance(E) team(E) sx_bfd(OE)
sx_netdev(OE) psample(E) sx_core(OE) 8021q(E) garp(E) mrp(E) mst_pciconf(OE) mst_pci(OE) xt_hl(E)
xt_tcpudp(E) ip6_tables(E) nft_chain_nat_ipv4(E) nf_nat_ipv4(E) nft_compat(E) nft_counter(E)
xt_contrack(E) nf_nat(E) jc42(E) nf_contrack_netlink(E) nf_contrack(E) nf_defrag_ipv6(E)
nf_defrag_ipv4(E) libcrc32c(E) xfrm_user(E) xfrm_algo(E) mlxsw_minimal(E) mlxsw_i2c(E) i2c_mux_reg(E)
i2c_mux(E) mlxreg_hotplug(E) mlxreg_io(E) i2c_mlxcp(E) leds_mlxreg(E) mei_wdt(E) evdev(E) intel_rapl(E)
x86_pkg_temp_thermal(E) intel_powerclamp(E) kvm_intel(E) kvm(E) mlx_platform(E) irqbypass(E)
crct10dif_pclmul(E) crc32_pclmul(E) ghash_clmulni_intel(E) intel_cstate(E)
[ 655.493833] intel_uncore(E) intel_rapl_perf(E) pcspkr(E) sg(E) iTCO_wdt(E) iTCO_vendor_support(E)
mei_me(E) mei(E) bonding(E) video(E) button(E) pcc_cpufreq(E) ebt_vlan(E) ebtable_broute(E) bridge(E) stp(E)
llc(E) ebtable_nat(E) ebtable_filter(E) ebtables(E) nf_tables(E) nfnetlink(E) xdpe12284(E) at24(E)
ledtrig_timer(E) tmp102(E) drm(E) lm75(E) coretemp(E) max1363(E) industrialio_triggered_buffer(E)
kfifo_buf(E) industrialio(E) fuse(E) tps53679(E) pmbus(E) pmbus_core(E) i2c_dev(E) configfs(E) ip_tables(E)
x_tables(E) autofs4(E) loop(E) ext4(E) crc16(E) mbcache(E) jbd2(E) crc32c_generic(E) fscrypto(E) ecb(E)
crypto_simd(E) cryptd(E) glue_helper(E) aes_x86_64(E) nvme(E) nvme_core(E) nls_utf8(E) nls_cp437(E)
nls_ascii(E) vfat(E) fat(E) overlay(E) squashfs(E) zstd_decompress(E) xxhash(E) sd_mod(E)
[ 656.337476] gpio_ich(E) ahci(E) mlxsw_core(E) libahci(E) devlink(E) crc32c_intel(E) libata(E) i2c_i801(E)
scsi_mod(E) lpc_ich(E) mfd_core(E) ehci_pci(E) ehci_hcd(E) usbcore(E) e1000e(E) usb_common(E) fan(E)
thermal(E) [ 656.569590] CR2: 0000000000000000
```



You can also specify a file name and number of lines in order to show the last number of lines.

- Example:

```
admin@sonic:~$ show kdump logging dmesg.202106242337 -l 20 [ 653.525427]
__handle_sysrq.cold.9+0x45/0xf2
[ 653.576487] write_sysrq_trigger+0x2b/0x30 [ 653.625472]
proc_reg_write+0x39/0x60
    [ 653.669252] vfs_write+0xa5/0x1a0
    [ 653.708881] ksys_write+0x57/0xd0
[ 653.748501] do_syscall_64+0x53/0x110
[ 653.792287] entry_SYSCALL_64_after_hwframe+0x44/0xa9
[ 653.852707] RIP: 0033:0x7fcfa27b504
[ 653.895452] Code: 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b3 0f 1f 80 00 00 00 00 48 8d 05 f9 61 0d 00 8b 00
85 c0 75 13 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 c3 0f 1f 00 41 54 49 89 d4 55 48 89 f5 53
[ 654.120195] RSP: 002b:00007ffe697690f8 EFLAGS: 00000246 ORIG_RAX:
0000000000000001
[ 654.210778] RAX: ffffffffda RBX: 0000000000000002 RCX: 00007fcfa27b504 [ 654.296157] RDX:
0000000000000002 RSI: 000055a6e4d1b3f0 RDI: 0000000000000001 [ 654.381543] RBP: 000055a6e4d1b3f0
R08: 000000000000000a R09: 00007fcfa2cc5f0 [ 654.466925] R10: 000000000000000a R11:
0000000000000246 R12: 00007fcfa34d760 [ 654.552310] R13: 0000000000000002 R14: 00007fcfa348760
R15: 0000000000000002 [ 654.637694] Modules linked in: binfmt_misc(E) nft_chain_route_ipv6(E)
nft_chain_route_ipv4(E) xt_TCPMSS(E) dummy(E) team_mode_loadbalance(E) team(E) sx_bfd(OE)
sx_netdev(OE) psample(E) sx_core(OE) 8021q(E) garp(E) mrp(E) mst_pciconf(OE) mst_pci(OE) xt_hl(E)
xt_tcpudp(E) ip6_tables(E) nft_chain_nat_ipv4(E) nf_nat_ipv4(E) nft_compat(E) nft_counter(E)
xt_conntrack(E) nf_nat(E) jc42(E) nf_conntrack_netlink(E) nf_conntrack(E) nf_defrag_ipv6(E)
nf_defrag_ipv4(E) libcrc32c(E) xfrm_user(E) xfrm_algo(E) mlxsw_minimal(E) mlxsw_i2c(E) i2c_mux_reg(E)
i2c_mux(E) mlxreg_hotplug(E) mlxreg_io(E) i2c_mlxcp(E) leds_mlxreg(E) mei_wdt(E) evdev(E) intel_rapl(E)
x86_pkg_temp_thermal(E) intel_powerclamp(E) kvm_intel(E) kvm(E) mlx_platform(E) irqbypass(E)
crct10dif_pclmul(E) crc32_pclmul(E) ghash_clmulni_intel(E) intel_cstate(E)
[ 655.493833] intel_uncore(E) intel_rapl_perf(E) pcspkr(E) sg(E) iTCO_wdt(E) iTCO_vendor_support(E)
mei_me(E) mei(E) bonding(E) video(E) button(E) pcc_cpufreq(E) ebt_vlan(E) ebt_table_broute(E) bridge(E) stp(E)
llc(E) ebt_table_nat(E) ebt_table_filter(E) ebt_tables(E) nf_tables(E) nfnetlink(E) xdpe12284(E) at24(E)
ledtrig_timer(E) tmp102(E) drm(E) lm75(E) coretemp(E) max1363(E) industrialio_triggered_buffer(E)
kfifo_buf(E) industrialio(E) fuse(E) tps53679(E) pmbus(E) pmbus_core(E) i2c_dev(E) configfs(E) ip_tables(E)
x_tables(E) autofs4(E) loop(E) ext4(E) crc16(E) mbcache(E) jbd2(E) crc32c_generic(E) fscrypto(E) ecb(E)
crypto_simd(E) cryptd(E) glue_helper(E) aes_x86_64(E) nvme(E) nvme_core(E) nls_utf8(E) nls_cp437(E)
nls_ascii(E) vfat(E) fat(E) overlay(E) squashfs(E) zstd_decompress(E) xxhash(E) sd_mod(E)
[ 656.337476] gpio_ich(E) ahci(E) mlxsw_core(E) libahci(E) devlink(E) crc32c_intel(E) libata(E) i2c_i801(E)
scsi_mod(E) lpc_ich(E) mfd_core(E) ehci_pci(E) ehci_hcd(E) usbcore(E) e1000e(E) usb_common(E) fan(E)
thermal(E) [ 656.569590] CR2: 0000000000000000
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.29.2 Linux Kernel Dump config commands config kdump

Administrative state of kdump is stored in ConfigDB.

The variable USE_KDUMP in the file /etc/default/kdump-tools is set to 0 to disable kdump, and set to 1 to enable kdump.

Since this command might require changing the kernel parameters to specify the amount of memory reserved for the capture kernel (the kernel parameters which are exported through /proc/cmdline), a reboot is necessary. The command displays a message showing that kdump functionality will be either enabled or disabled following the next reboot.

- Usage:

```
admin@sonic:~$ config kdump
```

Commands:

disable	Disable the KDUMP mechanism
enable	Enable the KDUMP mechanism
memory	Configure the memory for KDUMP mechanism
num_dumps	Configure the maximum dump files of KDUMP mechanism

[Go Back To Beginning of the document or Beginning of this section](#)

2.1.30 LLDP

2.1.30.1 LLDP show commands

show lldp table

This command displays the brief summary of all LLDP

neighbors. • Usage:

```
show lldp table
```

- Example:

```
admin@sonic:~$ show lldp table
```

Capability codes: (R) Router, (B) Bridge, (O) Other

LocalPort	RemoteDevice	RemotePortID	Capability RemotePortDescr
-----	-----	-----	-----
Ethernet112	T1-1	hundredGigE1/2	BR T0-2:hundredGigE1/29
Ethernet116	T1-2	hundredGigE1/2	BR T0-2:hundredGigE1/30
eth0	swtor-b2lab2-1610	GigabitEthernet 0/2	OBR
-----	-----	-----	-----

Total entries displayed: 3



show lldp neighbors

This command displays more details about all LLDP neighbors or only the neighbors connected to a specific interface.

- Usage:

```
show lldp neighbors <interface_name>
```

- Example1: To display all neighbors in all interfaces

```
admin@sonic:~$ show lldp neighbors
```

```
-----
LLDP neighbors:
-----
```

```
Interface:      eth0, via: LLDP, RID: 1, Time: 0 day, 12:21:21
```

```
Chassis:
```

```
ChassisID:      mac 00:01:e8:81:e3:45 swtor-
```

```
SysName:        b2lab2-1610
```

```
SysDescr:       Dell Force10 Networks Real Time Operating System Software.
```

```
All Rights Reserved.Build Time: Tue Sep 22 11:21:54 PDT 2015
```

```
TTL: Capability: 20
```

```
Capability:      Repeater, on
```

```
Capability:      Bridge, on
```

```
Port:            Router, on
```

```
PortID:
```

```
ifname GigabitEthernet 0/2
```

```
VLAN:            162, pvid: yes
```

```
-----
Interface:      Ethernet116, via: LLDP, RID: 3, Time: 0 day, 12:20:49
```

```
Chassis:
```

```
ChassisID:      mac 4c:76:25:e7:f0:c0 T1-2
```

```
SysName:        Debian GNU/Linux 8 (jessie) Linux 4.9.0-8-amd64 #1 SMP Debian
```

```
SysDescr:
```

```
4.9.110-3+deb9u6 (2015-12-19) x86_64
```

```
TTL: MgmtIP:    120 10.11.162.40
```

```
Capability:      Bridge, on
```

```
Capability:      Router, on Wlan,
```

```
Capability:      off Station, off
```

```
Capability:
```

```
Port: PortID:    local hundredGigE1/2
```

```
PortDescr:      T0-2:hundredGigE1/30
```



Optionally, you can specify an interface name in order to display only that particular interface

- Example2:

```
admin@sonic:~$ show lldp neighbors Ethernet112 show lldp
neighbors Ethernet112

-----

LLDP neighbors:
-----

Interface:      Ethernet112, via: LLDP, RID: 2, Time: 0 day, 19:24:17
Chassis:
  ChassisID:      mac 4c:76:25:e5:e6:c0 T1-1
  SysName:        Debian GNU/Linux 8 (jessie) Linux 4.9.0-8-amd64 #1 SMP Debian
  SysDescr:
4.9.110-3+deb9u6 (2015-12-19) x86_64
  TTL:
  MgmtIP:
  Capability:
  Capability:
  Capability:
  Capability:
Port:
  PortID:
  PortDescr:
```



2.1.31 Loading, Reloading And Saving Configuration

This section explains the commands that are used to load the configuration from either the ConfigDB or from the minigraph.

2.1.31.1 Loading configuration from JSON file config load

This command is used to load the configuration from a JSON file like the file which SONiC saves its configuration to, `/etc/sonic/config_db.json`. This command loads the configuration from the input file (if user specifies this optional filename, it will use that input file. Otherwise, it will use the default `/etc/sonic/config_db.json` file as the input file) into CONFIG_DB. The configuration present in the input file is applied on top of the already running configuration. This command does not flush the config DB before loading the new configuration (i.e., If the configuration present in the input file is same as the current running configuration, nothing happens). If the config present in the input file is not present in running configuration, it will be added. If the config present in the input file differs (when key matches) from that of the running configuration, it will be modified as per the new values for those keys.

When user specifies the optional argument “-y” or “-yes”, this command forces the loading without prompting the user for confirmation. If the argument is not specified, it prompts the user to confirm whether user really wants to load this configuration file.

- Usage:

```
config load [-y|--yes] [<filename>]
```

- Example:

```
admin@sonic:~$ sudo config load
Load config from the file /etc/sonic/config_db.json? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/config_db.json --write-to-db
```

2.1.31.2 Loading configuration from minigraph (XML) file config load_minigraph

This command is used to load the configuration from `/etc/sonic/minigraph.xml`. When users do not want to use configuration from `config_db.json`, they can copy the `minigraph.xml` configuration file to the device and load it using this command. This command restarts various services running in the device and it takes some time to complete the command.

NOTE: If the user had logged in using SSH, users might get disconnected and some configuration failures might happen which might be hard to recover. Users need to reconnect their SSH sessions after configuring the management IP address. It is recommended to execute this command from console port

NOTE: Management interface IP address and default route (or specific route) may require reconfiguration in case if those parameters are not part of the `minigraph.xml`.



When user specifies the optional argument “-y” or “-yes”, this command forces the loading without prompting the user for confirmation. If the argument is not specified, it prompts the user to confirm whether user really wants to load this configuration file.

When user specifies the optional argument “-n” or “-no-service-restart”, this command loads the configuration without restarting dependent services running on the device. One use case for this option is during boot time when config-setup service loads minigraph configuration and there is no services running on the device.

When user specifies the optional argument “-t” or “-traffic-shift-away”, this command executes TSA command at the end to ensure the device remains in maintenance after loading minigraph.

- Usage:

```
config load_minigraph [-y|--yes] [-n|--no-service-restart] [-t|--traffic-shift-away]
```

- Example:

```
admin@sonic:~$ sudo config load_minigraph Reload
config from minigraph? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/config_db.json --write-to-db
```

2.1.31.3 Reloading Configuration config reload

This command is used to clear current configuration and import new configuration from the input file or from /etc/sonic/config_db.json. This command shall stop all services before clearing the configuration and it then restarts those services.

This command restarts various services running in the device and it takes some time to complete the command.

NOTE: If the user had logged in using SSH, users **might get disconnected** depending upon the new management IP address. Users need to reconnect their SSH sessions. In general, it is recommended to execute this command from console port after disconnecting all SSH sessions to the device. When users do “config reload” the newly loaded config may have management IP address, or it may not have management IP address. If mgmtIP is there in the newly loaded config file, that mgmtIP might be same as previously configured value or it might be different. This difference in mgmtIP address values results in following possible behaviours.

Case1: Previously configured mgmtIP is same as newly loaded mgmtIP. The SSH session may not be affected at all, but it’s possible that there will be a brief interruption in the SSH session. But, assuming the client’s timeout value isn’t on the order of a couple of seconds, the session would most likely just resume again as soon as the interface is reconfigured and up with the same IP. Case2: Previously configured mgmtIP is different from newly loaded mgmtIP. Users will lose their SSH connections. Case3: Newly loaded config does not have any mgmtIP. Users will lose their SSH connections.

NOTE: Management interface IP address and default route (or specific route) may require reconfiguration in case if those parameters are not part of the minigraph.xml.

When user specifies the optional argument “-y” or “-yes”, this command forces the loading without prompting the user for confirmation. If the argument is not specified, it prompts the user to confirm whether user really wants to load this configuration file.



When user specifies the optional argument “-n” or “-no-service-restart”, this command clear and loads the configuration without restarting dependent services running on the device. One use case for this option is during boot time when config-setup service loads existing old configuration and there is no services running on the device.

When user specifies the optional argument “-f” or “-force”, this command ignores the system sanity checks. By default a list of sanity checks are performed and if one of the checks fail, the command will not execute. The sanity checks include ensuring the system status is not starting, all the essential services are up and swss is in ready state.

- Usage:

```
config reload [-y|--yes] [-l|--load-sysinfo] [<filename>] [-n|--no-service-restart] [-f|--force]
```

- Example:

```
admin@sonic:~$ sudo config reload
Clear current config and reload config from the file /etc/sonic/config_db.json? [y/N]: y
Running command: systemctl stop dhcp_relay Running
command: systemctl stop swss Running command:
systemctl stop snmp
Warning: Stopping snmp.service, but it can still be activated by: snmp.timer
Running command: systemctl stop lldp Running
command: systemctl stop pmon Running
command: systemctl stop bgp Running command:
systemctl stop teamd
Running command: /usr/local/bin/sonic-cfggen -H -k Force10-Z9100-C32 --write-to-db
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/config_db.json --write-to-db
Running command: systemctl restart hostname-config Running command: systemctl restart interfaces-config
Timeout, server 10.11.162.42 not responding.
```

When some sanity checks fail below error messages can be seen

```
admin@sonic:~$ sudo config reload -y
System is not up. Retry later or use -f to avoid system checks
```

```
admin@sonic:~$ sudo config reload -y
Relevant services are not up. Retry later or use -f to avoid system checks
```

```
admin@sonic:~$ sudo config reload -y
SwSS container is not ready. Retry later or use -f to avoid system checks
```

2.1.31.4 Loading Management Configuration

config load_mgmt_config

This command is used to reconfigure hostname and mgmt interface based on device description file. This command either uses the optional file specified as argument or looks for the file “/etc/sonic/device_desc.xml”. If the file does not exist or if the file does not have valid fields for “hostname” and “ManagementAddress” (or “ManagementAddressV6”), it fails.

When user specifies the optional argument “-y” or “-yes”, this command forces the loading without prompting the user for confirmation. If the argument is not specified, it prompts the user to confirm whether user really wants to load this configuration file.



- Usage:

```
config load_mgmt_config [-y|--yes] [<filename>]
```

- Example:

```
admin@sonic:~$ sudo config load_mgmt_config Reload
config from minigraph? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -M /etc/sonic/device_desc.xml --write-to-db
```

2.1.31.5 Saving Configuration to a File for Persistence config save

This command is to save the config DB configuration into the user-specified filename or into the default /etc/sonic/config_db.json. This saves the configuration into the disk which is available even after reboots. Saved file can be transferred to remote machines for debugging. If users wants to load the configuration from this new file at any point of time, they can use “config load” command and provide this newly generated file as input. If users wants this newly generated file to be used during reboot, they need to copy this file to /etc/sonic/config_db.json.

- Usage:

```
config save [-y|--yes] [<filename>]
```

- Example (Save configuration to /etc/sonic/config_db.json):

```
admin@sonic:~$ sudo config save -y
```

- Example (Save configuration to a specified file):

```
admin@sonic:~$ sudo config save -y /etc/sonic/config2.json
```

[Go Back To Beginning of the document or Beginning of this section](#)

2.1.32 Loopback Interfaces

2.1.32.1 Loopback show commands

Please check show ip interfaces

2.1.32.2 Loopback config commands

This sub-section explains how to create and delete loopback interfaces.

config interface loopback

This command is used to add or delete loopback interfaces. It is recommended to use loopback names in the format “Loopbackxxx”, where “xxx” is number of 1 to 3 digits. Ex: “Loopback11”.

- Usage:

```
config loopback (add | del) <loopback_name>
```

- Example (Create the loopback with name “Loopback11”):

```
admin@sonic:~$ sudo config loopback add Loopback11
```

[Go Back To Beginning of the document or Beginning of this section](#)



2.1.33 VRF Configuration

2.1.33.1 VRF show commands

show vrf

This command displays all vrfs configured on the system along with interface binding to the vrf. If vrf-name is also provided as part of the command, if the vrf is created it will display all interfaces binding to the vrf, if vrf is not created nothing will be displayed.

- Usage:

```
show vrf [<vrf_name>]
```

- Example:

```
admin@sonic:~$ show vrf
VRF          Interfaces
-----
default      Vlan20 Vlan100
Vrf-red      Loopback11
              Eth0.100
              Loopback100
Vrf-blue     Loopback102
              Ethernet0.10
              PortChannel101
```

2.1.33.2 VRF config commands

config vrf add

This command creates vrf in SONiC system with provided vrf-name. • Usage:

```
config vrf add <vrf-name>
```

Note: vrf-name should always start with keyword “Vrf”

config vrf del

This command deletes vrf with name vrf-name. • Usage:

```
config vrf del <vrf-name>
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.34 Management VRF

2.1.34.1 Management VRF Show commands

show mgmt-vrf

This command displays whether the management VRF is enabled or disabled. It also displays the details about the the links (eth0, mgmt, lo-m) that are related to management VRF.

- Usage:

```
show mgmt-vrf
```

- Example:

```
admin@sonic:~$ show mgmt-vrf

ManagementVRF : Enabled

Management VRF interfaces in Linux:
348: mgmt: <NOARP,MASTER,UP,LOWER_UP> mtu 65536 qdisc noqueue state UP mode DEFAULT
group default qlen 1000
    link/ether f2:2a:d9:bc:e8:f0 brd ff:ff:ff:ff:ff:ff
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master mgmt state UP mode
DEFAULT group default qlen 1000
    link/ether 4c:76:25:f4:f9:f3 brd ff:ff:ff:ff:ff:ff
350: lo-m: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue master mgmt state
UNKNOWN mode DEFAULT group default qlen 1000
    link/ether b2:4c:c6:f3:e9:92 brd ff:ff:ff:ff:ff:ff

NOTE: The management interface "eth0" shows the "master" as "mgmt" since it is part of management
VRF.
```

show mgmt-vrf routes

This command displays the routes that are present in the routing table 5000 that is meant for management VRF.

- Usage:

```
show mgmt-vrf routes
```

- Example:

```
admin@sonic:~$ show mgmt-vrf routes

Routes in Management VRF Routing Table: default via 10.16.210.254 dev eth0 metric 201
broadcast 10.16.210.0 dev eth0 proto kernel scope link src 10.16.210.75 10.16.210.0/24 dev eth0
proto kernel scope link src 10.16.210.75
local 10.16.210.75 dev eth0 proto kernel scope host src 10.16.210.75 broadcast 10.16.210.255 dev
eth0 proto kernel scope link src 10.16.210.75 broadcast 127.0.0.0 dev lo-m proto kernel scope link
src 127.0.0.1 127.0.0.0/8 dev lo-m proto kernel scope link src 127.0.0.1
local 127.0.0.1 dev lo-m proto kernel scope host src 127.0.0.1
broadcast 127.255.255.255 dev lo-m proto kernel scope link src 127.0.0.1
```



show management_interface address

This command displays the IP address(es) configured for the management interface “eth0” and the management network default gateway.

- Usage:

```
show management_interface address
```

- Example:

```
admin@sonic:~$ show management_interface address
Management IP address = 10.16.210.75/24
Management NetWork Default Gateway = 10.16.210.254
Management IP address = FC00:2::32/64
Management Network Default Gateway = fc00:2::1
```

show snmpagentaddress

This command displays the configured SNMP agent IP addresses. • Usage:

```
show snmpagentaddress
```

- Example:

```
admin@sonic:~$ show snmpagentaddress
ListenIP      ListenPort ---- ListenVrf
-----
1.2.3.4       787      mgmt
```

show snmptrap

This command displays the configured SNMP Trap server IP addresses. • Usage:

```
show snmptrap
```

- Example:

```
admin@sonic:~$ show snmptrap
Version  TrapReceiverIP      Port    VRF    Community
-----
2        31.31.31.31               456    mgmt    public
```



2.1.34.2 Management VRF Config commands

config vrf add mgmt

This command enables the management VRF in the system. This command restarts the “interfaces-config” service which in turn regenerates the /etc/network/interfaces file and restarts the “networking” service. This creates a new interface and l3mdev CGROUP with the name as “mgmt” and enslaves the management interface “eth0” into this master interface “mgmt”. Note that the VRFName “mgmt” (or “management”) is reserved for management VRF. i.e. Data VRFs should not use these reserved VRF names.

- Usage:

```
config vrf add mgmt
```

- Example:

```
admin@sonic:~$ sudo config vrf add mgmt
```

config vrf del mgmt

This command disables the management VRF in the system. This command restarts the “interfaces-config” service which in turn regenerates the /etc/network/interfaces file and restarts the “networking” service. This deletes the interface “mgmt” and deletes the l3mdev CGROUP named “mgmt” and puts back the management interface “eth0” into the default VRF. Note that the VRFName “mgmt” (or “management”) is reserved for management VRF. i.e. Data VRFs should not use these reserved VRF names.

- Usage:

```
config vrf del mgmt
```

- Example:

```
admin@sonic:~$ sudo config vrf del mgmt
```

config snmpagentaddress add

This command adds the SNMP agent IP address on which the SNMP agent is expected to listen. When SNMP agent is expected to work as part of management VRF, users should specify the optional vrf_name parameter as “mgmt”. This configuration goes into snmpd.conf that is used by SNMP agent. SNMP service is restarted to make this configuration effective in SNMP agent.

- Usage:

```
config snmpagentaddress add [-p <port_num>] [-v <vrf_name>] agentip
```

- Example:

```
admin@sonic:~$ sudo config snmpagentaddress add -v mgmt -p 123 21.22.13.14
```

Note: For this example, configuration goes into /etc/snmp/snmpd.conf inside snmp docker as follows. When “-v” parameter is not used, the additional “%” in the following line will not be present.

```
agentAddress 21.22.13.14:123%mgmt
```



config snmpagentaddress del

This command deletes the SNMP agent IP address on which the SNMP agent is expected to listen. When users had added the agent IP as part of “mgmt” VRF, users should specify the optional vrf_name parameter as “mgmt” while deleting as well. This configuration is removed from snmpd.conf that is used by SNMP agent. SNMP service is restarted to make this configuration effective in SNMP agent.

- Usage:

```
config snmpagentaddress del [-p <port_num>] [-v <vrf_name>] agentip
```

- Example:

```
admin@sonic:~$ sudo config snmpagentaddress del -v mgmt -p 123 21.22.13.14
```

config snmptrap modify

This command modifies the SNMP trap server IP address to which the SNMP agent is expected to send the traps. Users can configure one server IP address for each SNMP version to send the traps. When SNMP agent is expected to send traps as part of management VRF, users should specify the optional vrf_name parameter as “mgmt”. This configuration goes into snmpd.conf that is used by SNMP agent. SNMP service is restarted to make this configuration effective in SNMP agent.

- Usage:

```
config snmptrap modify <snmp_version> [-p <port_num>] [-v <vrf_name>] [-c <community>]  
trapserverip
```

- Example:

```
admin@sonic:~$ sudo config snmptrap modify 2 -p 456 -v mgmt 21.21.21.21
```

For this example, configuration goes into /etc/snmp/snmpd.conf inside snmp docker as follows. When “-v” parameter is not used, the additional “%” in the following line will not be present. In case of SNMPv1, “trapsink” will be updated, in case of v2, “trap2sink” will be updated and in case of v3, “informsink” will be updated.

```
trap2sink 31.31.31.31:456%mgmt public
```

config snmptrap del

This command deletes the SNMP Trap server IP address to which SNMP agent is expected to send TRAPs. When users had added the trap server IP as part of “mgmt” VRF, users should specify the optional vrf_name parameter as “mgmt” while deleting as well. This configuration is removed from snmpd.conf that is used by SNMP agent. SNMP service is restarted to make this configuration effective in SNMP agent.

- Usage:

```
config snmptrap del [-p <port_num>] [-v <vrf_name>] [-c <community>] trapserverip
```

- Example:

```
admin@sonic:~$ sudo config snmptrap del -v mgmt -p 123 21.22.13.14
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.35 Muxcable

2.1.35.1 Muxcable Show commands

show muxcable status

This command displays all the status of either all the ports which are connected to muxcable or any individual port selected by the user. The resultant table or json output will show the current status of muxcable on the port (auto/active) and also the health of the muxcable.

- Usage:

```
show muxcable status [OPTIONS] [PORT]
```

While displaying the muxcable status, users can configure the following fields

- PORT optional - Port name should be a valid port
- -json optional - option to display the result in json format. By default output will be in tabular format.

With no optional argument, all the ports muxcable status will be displayed in tabular form, or user can pass -json option to display in json format

- Example:

```
admin@sonic:~$ show muxcable status
PORT          STATUS    HEALTH
-----
Ethernet32    active    HEALTHY
Ethernet0     auto      HEALTHY
```

```
admin@sonic:~$ show muxcable status --json
{
  "MUX_CABLE":
  {
    "Ethernet32":
      {"STATUS": "active", "HEALTH": "HEALTHY"},
    "Ethernet0":
      {"STATUS": "auto", "HEALTH": "HEALTHY"}
  }
}
```

```
admin@sonic:~$ show muxcable status Ethernet0
PORT          STATUS    HEALTH
-----
Ethernet0     auto      HEALTHY
```

```
admin@sonic:~$ show muxcable status Ethernet0 --json
{
  "MUX_CABLE":
  {
    "Ethernet0":
      {
        "STATUS": "auto",
        "HEALTH": "HEALTHY"
      }
  }
}
```



show muxcable config

This command displays all the configurations of either all the ports which are connected to muxcable or any individual port selected by the user. The resultant table or json output will show the current configurations of muxcable on the port(active/standby) and also the ipv4 and ipv6 address of the port as well as peer TOR ip address with the hostname.

- Usage:

```
show muxcable config [OPTIONS] [PORT]
```

With no optional argument, all the ports muxcable configuration will be displayed in tabular form. While displaying the muxcable configuration, users can configure the following fields

- PORT optional - Port name should be a valid port
- -json optional - option to display the result in json format. By default output will be in tabular format.
- Example:

```
admin@sonic:~$ show muxcable config
SWITCH_NAME    PEER_TOR
-----
sonic          10.1.1.1

port           state      ipv4       ipv6
-----
Ethernet0      active     10.1.1.1   fc00::75
```

```
admin@sonic:~$ show muxcable config --json
{"MUX_CABLE":
  {"PEER_TOR": "10.1.1.1", "PORTS":
    {"Ethernet0":
      {
        "STATE": "active",
        "SERVER": {"IPv4": "10.1.1.1", "IPv6": "fc00::75"}
      }
    }
  }
}
```




```
admin@sonic:~$ show muxcable config Ethernet0
```

SWITCH_NAME	PEER_TOR
sonic	10.1.1.1

port	state	ipv4	ipv6
Ethernet0	active	10.1.1.1	fc00::75

```
admin@sonic:~$ show muxcable config Ethernet0 --json
{ "MUX_CABLE":
  { "PORTS":
    { "Ethernet0":
      { "STATE": "active", "SERVER":
        { "IPv4": "10.1.1.1", "IPv6": "fc00::75" }
      }
    }
  }
}
```

show muxcable ber-info

This command displays the ber(Bit error rate) of the port user provides on the target user provides. The target provided as an integer corresponds to actual target as. 0 -> local 1 -> tor 1 2 -> tor 2 3 -> nic

- Usage:

```
Usage: show muxcable ber-info [OPTIONS] PORT TARGET
```

- PORT required - Port number should be a valid port
- TARGET required - the actual target to get the ber info

- of. • Example:

```
admin@sonic:~$ show muxcable ber-info 1 1
```

Lane1	Lane2
0	0

show muxcable eye-info

This command displays the eye info in mv(milli volts) of the port user provides on the target user provides. The target provided as an integer corresponds to actual target as. 0 -> local 1 -> tor 1 2 -> tor 2 3 -> nic

- Usage:

```
Usage: show muxcable eye-info [OPTIONS] PORT TARGET
```

- PORT required - Port number should be a valid port
- TARGET required - the actual target to get the eye info of.

- Example:

```
admin@sonic:~$ show muxcable eye-info 1 1
```

Lane1	Lane2
632	622



show muxcable health

This command displays the hardware health of the Y-cable which are connected to muxcable. The resultant table or json output will show the current hardware health of the cable as Ok, Not Ok, Unknown.

- Usage:

```
show muxcable health [OPTIONS] [PORT]
```

While displaying the muxcable health, users need to provide the following fields

- PORT required - Port name should be a valid port
- -json optional - - option to display the result in json format. By default output will be in tabular format.

-Ok means the cable is healthy

in order to determine whether the health of the cable is Ok the following are checked

- the vendor name is correct able to be read
- the FW is correctly loaded for SerDes by reading the appropriate register val
- the Counters for UART are displaying healthy status i.e Error Counters , retry Counters for UART or internal xfer protocols are below a threshold
- Example:

```
admin@sonic:~$ show muxcable health Ethernet4
PORT      ATTR    HEALTH
-----
Ethernet4  health  Ok
```

```
admin@sonic:~$ show muxcable health Ethernet4 --json
```

```
{"health": "Ok" }
```

show muxcable queueinfo

This command displays the queue info of the Y-cable which are connected to muxcable. The resultant table or json output will show the queue info in terms transactions for the UART stats in particular currently relevant to the MCU of the cable.

- Usage:

```
show muxcable queueinfo [OPTIONS] [PORT]
```



While displaying the muxcable queueinfo, users need to provide the following fields

- PORT required - Port name should be a valid port
- -json optional - - option to display the result in json format. By default output will be in tabular format.

the result will be displayed like this, each item in the dictionary shows the health of the attribute in the queue

```
"{'VSC': {'r_ptr': 0, 'w_ptr': 0, 'total_count': 0, 'free_count': 0, 'buff_addr': 0, 'node_size': 0}, 'UART1': {'r_ptr': 0, 'w_ptr': 0, 'total_count': 0, 'free_count': 0, 'buff_addr': 209870, 'node_size': 1682183}, 'UART2': {'r_ptr': 13262, 'w_ptr': 3, 'total_count': 0, 'free_count': 0, 'buff_addr': 12, 'node_size': 0}}
```

- Example:

```
admin@sonic:~$ show muxcable queueinfo Ethernet0
```

PORT	ATTR	VALUE
-----	-----	-----
Ethernet0	uart_stat1	2
Ethernet0	uart_stat2	1

```
admin@sonic:~$ show muxcable queueinfo Ethernet4 --json
{
  "uart_stat1": "2",
  "uart_stat2": "1",
}
```

show muxcable operationtime

This command displays the operationtime of the Y-cable which are connected to muxcable. The resultant table or json output will show the current operation time of the cable as hh:mm:ss format. Operation time means the time since the last time the reseated/reset of the cable is done, and the time would be in the format specified

- Usage:

```
show muxcable operationtime [OPTIONS] [PORT]
```

While displaying the muxcable operationtime, users need to provide the following fields

- PORT required - Port name should be a valid port
- -json optional - - option to display the result in json format. By default output will be in tabular format.
- Example:

```
admin@sonic:~$ show muxcable operationtime Ethernet4
```

PORT	ATTR	OPERATION_TIME
-----	-----	-----
Ethernet4	operation_time	00:22:22

```
admin@sonic:~$ show muxcable operationtime Ethernet4 --json
{ "operation_time": "00:22:22" }
```



show muxcable resetcause

This command displays the resetcause of the Y-cable which are connected to muxcable. The resultant table or json output will show the most recent reset cause of the cable as string format.

- Usage:

```
show muxcable resetcause [OPTIONS] [PORT]
```

While displaying the muxcable resetcause, users need to provide the following fields

- PORT required - Port name should be a valid port
- -json optional - option to display the result in json format. By default output will be in tabular format.

the reset cause only records NIC MCU reset status. The NIC MCU will automatically broadcast the reset cause status to each TORs, corresponding values returned display cold reset if the last reset is cold reset (ex. HW/SW reset, power reset the cable, or reboot the NIC server) display warm reset if the last reset is warm reset (ex. sudo config mux firmware activate....) the value is persistent, no clear on read

- Example:

```
admin@sonic:~$ show muxcable resetcause Ethernet4
PORT      ATTR      RESETCAUSE
-----
Ethernet4  reset_cause  warm reset
```

```
admin@sonic:~$ show muxcable resetcause Ethernet4 --json
{ "reset_cause": "warm reset" }
```

2.1.35.2 Muxcable Config commands

config muxcable mode

This command is used for setting the configuration of a muxcable Port/all ports to be active or auto. The user has to enter a port number or else all to make the muxcable config operation on all the ports. Depending on the status of the muxcable port state the resultant output could be OK or INPROGRESS . OK would imply no change on the state, INPROGRESS would mean the toggle is happening in the background.

- Usage:

```
config muxcable mode [OPTIONS] <operation_status> <port_name>
```

While configuring the muxcable, users needs to configure the following fields for the operation

- <auto/active> operation_state, permitted operation to be configured which can only be auto or active
- PORT optional - Port name should be a valid port
- -json optional - option to display the result in json format. By default output will be in tabular format.



- Example:

```
admin@sonic:~$ sudo config muxcable mode active Ethernet0
```

port	state
-----	-----
Ethernet0	OK

```
admin@sonic:~$ sudo config muxcable mode --json active Ethernet0
{ "Ethernet0": "OK" }
```

```
admin@sonic:~$ sudo config muxcable mode active all
```

port	state
-----	-----
Ethernet0	OK
Ethernet32	INPROGRESS

```
admin@sonic:~$ sudo config muxcable mode active all --json
{
  "Ethernet32": "INPROGRESS",
  "Ethernet0": "OK"
}
```

config muxcable prbs enable/disable

This command is used for setting the configuration and enable/disable of prbs on a port user provides. While enabling in addition to port the user also needs to provide the target, prbs mode and lane map on which the user intends to run prbs on. The target reflects where the enable/disable will happen.

- Usage:

```
config muxcable prbs enable [OPTIONS] PORT TARGET MODE_VALUE LANE_MAP config
muxcable prbs disable [OPTIONS] PORT TARGET
```

While configuring the muxcable, users need to configure the following fields for the operation

- PORT required - Port number should be a valid port
- TARGET required - the actual target to run the prbs on 0 -> local side, 1 -> TOR 1 2 -> TOR 2 3 -> NIC
- MODE_VALUE required - the mode/type for configuring the PRBS mode. 0x00 = PRBS 9, 0x01 = PRBS 15, 0x02 = PRBS 23, 0x03 = PRBS 31
- LANE_MAP required - an integer representing the lane_map to be run PRBS on 0bit for lane 0, 1bit for lane1 and so on. for example 3 -> 0b'0011, means running on lane0 and lane1
- Example:

```
admin@sonic:~$ sudo config muxcable prbs enable 1 1 3 3 PRBS config successful
admin@sonic:~$ sudo config muxcable prbs disable 1 0 PRBS disable successful
```



config muxcable loopback enable/disable

This command is used for setting the configuration and enable/disable of loopback on a port user provides. While enabling in addition to port the user also needs to provides the target and lane map on which the user intends to run loopback on. The target reflects where the enable/dsiable will happen.

- Usage:

```
config muxcable loopback enable [OPTIONS] PORT TARGET LANE_MAP config
muxcable loopback disable [OPTIONS] PORT TARGET
```

While configuring the muxcable, users needs to configure the following fields for the operation

- PORT required - Port number should be a valid port
- TARGET required - the actual target to run the loopback on 0 -> local side, 1 -> TOR 1 2 -> TOR 2 3 -> NIC
- LANE_MAP required - an integer representing the lane_map to be run loopback on 0bit for lane 0, 1bit for lane1 and so on. for example 3 -> 0b'0011 , means running on lane0 and lane1
- Example:

```
admin@sonic:~$ sudo config muxcable loopback enable 1 1 3 loopback config sucessful
admin@sonic:~$ sudo config muxcable loopback disable 1 0 loopback disable sucessfull
```

Go Back To Beginning of the document or Beginning of this section

2.1.36 Mirroring

2.1.36.1 Mirroring Show commands

show mirror_session

This command displays all the mirror sessions that are configured.

- Usage:

```
show mirror_session
```

- Example:

```
admin@sonic:~$ show mirror_session ERSPAN
Sessions
Name          Status SRC IPDST IP      GRE      DSCP      TTL      Queue  Policer Monitor Port
SRC Port  Direction
-----
-----
```



```
-----everflow0 active
10.1.0.32 10.0.0.7
```

SPAN Sessions

Name	Status	DST Port	SRC Port	Direction
port0	active	Ethernet0	PortChannel10	rx

2.1.36.2 Mirroring Config commands

config mirror_session

This command is used to add or remove mirroring sessions. Mirror session is identified by “session_name”. This command supports configuring both SPAN/ERSPAN sessions. In SPAN user can configure mirroring of list of source ports/LAG to destination port in ingress/egress/both directions. In ERSPAN user can configure mirroring of list of source ports/LAG to a destination IP. Both SPAN/ERSPAN support ACL based mirroring and can be used in ACL configurations.

While adding a new ERSPAN session, users need to configure the following fields that are used while forwarding the mirrored packets.

1. source IP address,
2. destination IP address,
3. DSCP (QoS) value with which mirrored packets are forwarded
4. TTL value
5. optional - GRE Type in case if user wants to send the packet via GRE tunnel. GRE type could be anything; it could also be left as empty; by default, it is 0x8949 for Mellanox; and 0x88be for the rest of the chips.
6. optional - Queue in which packets shall be sent out of the device. Valid values 0 to 7 for most of the devices. Users need to know their device and the number of queues supported in that device.
7. optional - Policer which will be used to control the rate at which frames are mirrored.
8. optional - List of source ports which can have both Ethernet and LAG ports.
9. optional - Direction - Mirror session direction when configured along with Source port. (Supported rx/tx/both. default direction is both)

• Usage:

```
config mirror_session erspan add <session_name> <src_ip> <dst_ip> <dscp> <ttl> [gre_type] [queue] [policer
<policer_name>] [source-port-list] [direction]
```

The following command is also supported to be backward compatible. This command will be deprecated in future releases.

```
config mirror_session add <session_name> <src_ip> <dst_ip> <dscp> <ttl> [gre_type] [queue]
```

• Example:

```
root@T1-2:~# config mirror_session add mrr_legacy 1.2.3.4 20.21.22.23 8 100 0x6558 0
root@T1-2:~# show mirror_session
```

Name	Status	SRC IP	DST IP	GRE	DSCP	TTL	Queue
Policer	Monitor Port	SRC Port	Direction				



```

-----
-----
mrr_legacy  inactive  1.2.3.4          20.21.22.23  0x6558  8          100      0

root@T1-2:~# config mirror_session erspan add mrr_abcd 1.2.3.4 20.21.22.23 8 100 0x6558 0
root@T1-2:~# show mirror_session
Name          Status      SRC IP      DST IP      GRE      DSCP      TTL      Queue
Policer       Monitor Port SRC Port    Direction
-----
-----
mrr_abcd      inactive  1.2.3.4          20.21.22.23  0x6558  8          100      0
root@T1-2:~#

root@T1-2:~# config mirror_session erspan add mrr_port 1.2.3.4 20.21.22.23 8 100 0x6558 0 Ethernet0
root@T1-2:~# show mirror_session
Name          Status      SRC IP      DST IP      GRE      DSCP      TTL      Queue
Policer       Monitor Port SRC Port    Direction
-----
-----
mrr_port      inactive  1.2.3.4          20.21.22.23  0x6558  8          100      0
Ethernet0     both
root@T1-2:~#

```

While adding a new SPAN session, users need to configure the following fields that are used while forwarding the mirrored packets.

1. destination port,
2. optional - List of source ports- List of source ports which can have both Ethernet and LAG ports.
3. optional - Direction - Mirror session direction when configured along with Source port. (Supported rx/tx/both. default direction is both)
4. optional - Queue in which packets shall be sent out of the device. Valid values 0 to 7 for most of the devices. Users need to know their device and the number of queues supported in that device.
5. optional - Policer which will be used to control the rate at which frames are mirrored.

- Usage:

```
config mirror_session span add <session_name> <dst_port> [source-port-list] [direction] [queue]
[policer <policer_name>]
```

- Example:

```

root@T1-2:~# config mirror_session span add port0 Ethernet0
Ethernet4,PortChannel001,Ethernet8
root@T1-2:~# show mirror_session
Name      Status      DST Port      SRC Port      Direction
-----
-----
port0     active      Ethernet0     Ethernet4,PortChannel10,Ethernet8  both
root@T1-2:~#

```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.1.37 NAT

2.1.37.1 NAT Show commands

show nat config

This command displays the NAT configuration.

- Usage:

```
show nat config [static | pool | bindings | globalvalues | zones]
```

With no optional arguments, the whole NAT configuration is displayed.

- Example:

```
admin@sonic:~$ show nat config static
```

Nat Type	IP Protocol	Global IP	Global L4 Port	Local IP	Local L4 Port	Twice-Nat Id
dnat	all	65.55.45.5	---	10.0.0.1	---	
-						
dnat	all	65.55.45.6	---	10.0.0.2	---	
-						
dnat	tcp	65.55.45.7	2000	20.0.0.1	4500	
snat	tcp	20.0.0.2	4000	65.55.45.8	1030	
1						

```
admin@sonic:~$ show nat config pool
```

Pool Name	Global IP Range	Global L4 Port Range
Pool1	65.55.45.5	1024-65535
Pool2	65.55.45.6-65.55.45.8	---
Pool3	65.55.45.10-65.55.45.15	500-1000

```
admin@sonic:~$ show nat config bindings
```

Binding Name	Pool Name	Access-List	Nat Type	Twice-Nat Id
-----	-----	-----	-----snat	-----
Bind1	Pool1	1	snat	1
Bind2	Pool2	2	snat	--
Bind3	Pool3			

```
admin@sonic:~$ show nat config globalvalues
```

```
Admin Mode      : enabled
Global Timeout  : 600 secs
TCP Timeout     : 86400 secs
UDP Timeout     : 300 secs
```

```
admin@sonic:~$ show nat config zones
```

Port	Zone
----	----
Ethernet2	0
Vlan100	1



show nat statistics

This command displays the NAT translation statistics for each entry.

- Usage:

```
show nat statistics
```

- Example:

```
admin@sonic:~$ show nat statistics
```

Protocol	Source	Destination	Packets	Bytes
all	10.0.0.1	---	802	1009280
all	10.0.0.2	---	23	5590
tcp	20.0.0.1:4500	---	110	12460
udp	20.0.0.1:4000	---	1156	789028
tcp	20.0.0.1:6000	---	30	34800
tcp	20.0.0.1:5000	65.55.42.1:2000	128	110204
tcp	20.0.0.1:5500	65.55.42.1:2000	8	3806

show nat translations

This command displays the NAT translation entries.

- Usage:

```
show nat translations [count]
```

Giving the optional count argument displays only the details about the number of translation entries.

- Example:

```
admin@sonic:~$ show nat translations
```

Static NAT Entries		4
Static NAPT Entries		2
Dynamic NAT Entries		0
Dynamic NAPT Entries		4
Static Twice NAT Entries		0
Static Twice NAPT Entries		4
Dynamic Twice NAT Entries		0
Dynamic Twice NAPT Entries		0
Total SNAT/SNAPT Entries		9
Total DNAT/DNAPT Entries		9
Total Entries		14

Protocol	Source	Destination	Translated Source	Translated
all	10.0.0.1	---	65.55.42.2	---
all	---	65.55.42.2	---	10.0.0.1
all	10.0.0.2	---	65.55.42.3	---
all	---	65.55.42.3	---	10.0.0.2
tcp	20.0.0.1:4500	---	65.55.42.1:2000	---
tcp	---	65.55.42.1:2000	---	20.0.0.1:4500
udp	20.0.0.1:4000	---	65.55.42.1:1030	---
udp	---	65.55.42.1:1030	---	20.0.0.1:4000
tcp	20.0.0.1:6000	---	65.55.42.1:1024	---
tcp	---	65.55.42.1:1024	---	20.0.0.1:6000
tcp	20.0.0.1:5000	65.55.42.1:2000	65.55.42.1:1025	20.0.0.1:4500
tcp	20.0.0.1:4500	65.55.42.1:1025	65.55.42.1:2000	20.0.0.1:5000
tcp	20.0.0.1:5500	65.55.42.1:2000	65.55.42.1:1026	20.0.0.1:4500
tcp	20.0.0.1:4500	65.55.42.1:1026	65.55.42.1:2000	20.0.0.1:5500

```
admin@sonic:~$ show nat translations count
```

Static NAT Entries		4
Static NAPT Entries		2
Dynamic NAT Entries		0
Dynamic NAPT Entries		4
Static Twice NAT Entries		0
Static Twice NAPT Entries		4
Dynamic Twice NAT Entries		0
Dynamic Twice NAPT Entries		0
Total SNAT/SNAPT Entries		9
Total DNAT/DNAPT Entries		9
Total Entries		14





2.1.37.2 NAT Config commands

config nat add static

This command is used to add a static NAT or NAPT entry. When configuring the Static NAT entry, user has to specify the following fields with 'basic' keyword.

1. Global IP address,
2. Local IP address,
3. NAT type (snat / dnat) to be applied on the Global IP address. Default value is dnat. This is optional argument.
4. Twice NAT Id. This is optional argument used in case of twice nat configuration.

When configuring the Static NAPT entry, user has to specify the following fields.

1. IP protocol type (tcp / udp)
2. Global IP address + Port
3. Local IP address + Port
4. NAT type (snat / dnat) to be applied on the Global IP address + Port. Default value is dnat. This is optional argument.
5. Twice NAT Id. This is optional argument used in case of twice nat configuration.

- Usage:

```
config nat add static {{basic (global-ip) (local-ip)} | {{tcp | udp} (global-ip) (global-port) (local-ip) (local-port)}}
[-nat_type {snat | dnat}] [-twice_nat_id (value)]
```

To delete a static NAT or NAPT entry, use the command below. Giving the all argument deletes all the configured static NAT and NAPT entries.

```
config nat remove static {{basic (global-ip) (local-ip)} | {{tcp | udp} (global-ip) (global-port) (local-ip) (local-port)}} | all}
```

- Example:

```
admin@sonic:~$ sudo config nat add static basic 65.55.45.1 12.12.12.14 -nat_type
dnat
admin@sonic:~$ sudo config nat add static tcp 65.55.45.2 100 12.12.12.15 200 -
nat_type dnat
```

```
admin@sonic:~$ show nat translations
```

```
Static NAT Entries      ..... 2
Static NAPT Entries     ..... 2
Dynamic NAT Entries     ..... 0
Dynamic NAPT Entries    ..... 0
Static Twice NAT Entries ..... 0
Static Twice NAPT Entries ..... 0
Dynamic Twice NAT Entries ..... 0
Dynamic Twice NAPT Entries ..... 0
Total SNAT/SNAPT Entries ..... 2
Total DNAT/DNAPT Entries ..... 2
Total Entries           ..... 4

Protocol Source      Destination      Translated Source  Translated
Destination
-----
-----
all      12.12.12.14      ---      65.55.42.1      ---
all      ---      65.55.42.1      ---      12.12.12.14
tcp      12.12.12.15:200  ---      65.55.42.2:100  ---
tcp      ---      65.55.42.2:100  ---      12.12.12.15:200
```





config nat add pool

This command is used to create a NAT pool used for dynamic Source NAT or NATPT translations. Pool can be configured in one of the following combinations.

1. Global IP address range (or)
2. Global IP address + L4 port range (or)
3. Global IP address range + L4 port range.

- Usage:

```
config nat add pool (pool-name) (global-ip-range) (global-port-range)
```

To delete a NAT pool, use the command. Pool cannot be removed if it is referenced by a NAT binding. Giving the pools argument removes all the configured pools.

```
config nat remove {pool (pool-name) | pools}
```

- Example:

```
admin@sonic:~$ sudo config nat add pool pool1 65.55.45.2-65.55.45.10
admin@sonic:~$ sudo config nat add pool pool2 65.55.45.3 100-1024

admin@sonic:~$ show nat config pool
```

Pool Name	Global IP Range	Global Port Range
pool1	65.55.45.2-65.55.45.10	---
pool2	65.55.45.3	100-1024



config nat add binding

This command is used to create a NAT binding between a pool and an ACL. The following fields are needed for configuring the binding.

1. ACL is an optional argument. If ACL argument is not given, the NAT binding is applicable to match all traffic.
2. NAT type is an optional argument. Only DNAT type is supported for binding.
3. Twice NAT Id is an optional argument. This Id is used to form a twice nat grouping with the static NAT/NAPT entry configured with the same Id.

- Usage:

```
config nat add binding (binding-name) [(pool-name)] [(acl-name)] [-nat_type {snat | dnat}] [-twice_nat_id (value)]
```

To delete a NAT binding, use the command below. Giving the bindings argument removes all the configured bindings.

```
config nat remove {binding (binding-name) | bindings}
```

- Example:

```
admin@sonic:~$ sudo config nat add binding bind1 pool1 acl1
admin@sonic:~$ sudo config nat add binding bind2 pool2
admin@sonic:~$ show nat config bindings
```

Binding Name	Pool Name	Access-List	Nat Type	Twice-NAT Id
-----	-----	-----	-----	-----
bind1	pool1	acl1	snat	---
bind2	pool2		snat	---

config nat add interface

This command is used to configure NAT zone on an L3 interface. Default value of NAT zone on an L3 interface is 0. Valid range of zone values is 0-3.

- Usage:

```
config nat add interface (interface-name) -nat_zone (value)
```

To reset the NAT zone on an interface, use the command below. Giving the interfaces argument resets the NAT zone on all the L3 interfaces to 0.

```
config nat remove {interface (interface-name) | interfaces}
```

- Example:

```
admin@sonic:~$ sudo config nat add interface Ethernet28 -nat_zone 1
admin@sonic:~$ show nat config zones
```

Port	Zone
-----	-----
Ethernet0	0
Ethernet28	1
Ethernet22	0
Vlan2091	0



config nat set

This command is used to set the NAT timeout values. Different timeout values can be configured for the NAT entry timeout, NAPT TCP entry timeout, NAPT UDP entry timeout. Range for Global NAT entry timeout is 300 sec to 432000 sec, default value is 600 sec. Range for TCP NAT/NAPT entry timeout is 300 sec to 432000 sec, default value is 86400 sec. Range for UDP NAT/NAPT entry timeout is 120 sec to 600 sec, default value is 300 sec.

- Usage:

```
config nat set {tcp-timeout (value) | timeout (value) | udp-timeout (value)}
```

To reset the timeout values to the default values, use the command

```
config nat reset {tcp-timeout | timeout | udp-timeout}
```

- Example:

```
admin@sonic:~$ sudo config nat add set tcp-timeout 3600
admin@sonic:~$ show nat config globalvalues
Admin Mode       : enabled
Global Timeout   : 600 secs
TCP Timeout      : 600 secs
UDP Timeout      : 300 secs
```

config nat feature

This command is used to enable or disable the NAT feature.

- Usage:

```
config nat feature {enable | disable}
```

- Example:

```
admin@sonic:~$ sudo config nat feature enable
admin@sonic:~$ sudo config nat feature disable
```

2.1.37.3 NAT Clear commands

sonic-clear nat translations

This command is used to clear the dynamic NAT and NAPT translation entries.

- Usage:

```
sonic-clear nat translations
```

sonic-clear nat statistics

This command is used to clear the statistics of all the NAT and NAPT entries.

- Usage:

```
sonic-clear nat statistics
```

Go Back To Beginning of the document or Beginning of this section



2.1.38 NTP

2.1.38.1 NTP show commands

show ntp

This command displays a list of NTP peers known to the server as well as a summary of their state.

- Usage:

```
show ntp
```

- Example:

```
admin@sonic:~$ show ntp
synchronised to NTP server (204.2.134.164) at stratum 3
time correct to within 326797 ms
polling server every 1024 s
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
23.92.29.245	.XFAC.	16	u	-	1024	0	0.000	0.000	0.000
*204.2.134.164	46.233.231.73	2	u	916	1024	377	3.079	0.394	0.128

2.1.38.2 NTP Config Commands

This sub-section of commands is used to add or remove the configured NTP servers.

config ntp add

This command is used to add a NTP server IP address to the NTP server list. Note that more than one NTP server IP address can be added in the device.

- Usage:

```
config ntp add <ip_address>
```

- Example:

```
admin@sonic:~$ sudo config ntp add 9.9.9.9 NTP server 9.9.9.9 added to configuration Restarting ntp-config service...
```

2.2 PFC Watchdog Commands

Detailed description of the PFC Watchdog can be found on this [wiki page](#)

config pfcwd start <arguments>

This command starts PFC Watchdog

- Usage:

```
config pfcwd start --action drop all 400 --restoration-time 400 config pfcwd start --action forward Ethernet0 Ethernet8 400
```

config pfcwd stop

This command stops PFC Watchdog

- Usage:

```
config pfcwd stop
```

config pfcwd interval <interval_in_ms>

This command sets PFC Watchdog counter polling interval (in ms)

- Usage:

```
config pfcwd interval 200
```

config pfcwd counter_poll <enable/disable>

This command enables or disables PFCWD related counters polling

- Usage:

```
config pfcwd counter_poll disable
```



config pfcwd big_red_switch <enable/disable>

This command enables or disables PFCWD's "BIG RED SWITCH"(BRS). After enabling BRS PFC Watchdog will be activated on all ports/queues it is configured for no matter whether the storm was detected or not

- Usage:

```
config pfcwd big_red_switch enable
```

config pfcwd start_default

This command starts PFC Watchdog with the default settings.

- Usage:

```
config pfcwd start_default
```

Default values are the following:

- detection time - 200ms
- restoration time - 200ms
- polling interval - 200ms
- action - 'drop'

Additionally if number of ports in the system exceeds 32, all times will be multiplied by roughly $\frac{\text{num_ports}}{32}$.

show pfcwd config

This command shows current PFC Watchdog configuration

- Usage:

```
show pfcwd config
```

show pfcwd stats

This command shows current PFC Watchdog statistics (storms detected, packets dropped, etc)

- Usage:

```
show pfcwd stats
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.2.1 Platform Component Firmware

2.2.1.1 Platform Component Firmware show commands

show platform firmware status

This command displays platform components firmware status information.

- Usage:

```
show platform firmware status
```

- Example:

```
admin@sonic:~$ sudo show platform firmware status
```

Chassis	Module	Component	Version	Description
MSN3800	N/A	ONIE	2020.11-5.2.0022-9600	ONIE - Open Network
Install Environment		SSD	0202-000	SSD - Solid-State
Drive		BIOS	0AC1H004_02.02.008_9600	BIOS - Basic Input/
Output System		CPLD1	CPLD000120_REV0900	CPLD - Complex
Programmable Logic Device		CPLD2	CPLD000165_REV0500	CPLD - Complex
Programmable Logic Device		CPLD3	CPLD000166_REV0300	CPLD - Complex
Programmable Logic Device		CPLD4	CPLD000167_REV0100	CPLD - Complex
Programmable Logic Device				

show platform firmware updates

This command displays platform components firmware updates

information. • Usage:

```
show platform firmware updates [-i|--image]
```

- Options:

- `-i|--image`: show updates using current/next SONiC image

Valid values:

- current
- next

Default:

- current

- Example:

```
admin@sonic:~$ sudo show platform firmware updates
```



- Note:
 - current/next values for `-i|-image` are taken from `sonic-installer list`

show platform firmware version

This command displays platform components firmware utility version.

- Usage:


```
show platform firmware version
```
- Example:


```
admin@sonic:~$ show platform firmware version fwutil
version 2.0.0.0
```

2.2.1.2 Platform Component Firmware config commands

config platform firmware install

This command is used to install a platform component firmware. Both modular and non modular chassis platforms are supported.

- Usage:


```
config platform firmware install chassis component <component_name> fw <fw_path> [-y|--yes]
config platform firmware install module <module_name> component <component_name> fw <fw_path> [-y|--yes]
```
- Options:
 - `-y|--yes`: automatic yes to prompts. Assume “yes” as answer to all prompts and run non-interactively
- Example:


```
admin@sonic:~$ sudo config platform firmware install chassis component BIOS fw /
usr/local/lib/firmware/mellanox/sn3800/chassis1/bios.bin
Warning: Immediate cold reboot is required to complete BIOS firmware update. New firmware will be
installed, continue? [y/N]: y
Installing firmware: /usr/local/lib/firmware/mellanox/sn3800/chassis1/bios.bin

admin@sonic:~$ sudo config platform firmware install module Module1 component BIOS fw
https://www.mellanox.com/fw/sn3800/module1/bios.bin
Warning: Immediate cold reboot is required to complete BIOS firmware update. New firmware will be
installed, continue? [y/N]: y
Downloading firmware: [#####]
100%
Installing firmware:
/tmp/bios.bin
```

- Note:
 - `<fw_path>` can be absolute path or URL



config platform firmware update

This command is used to update a platform component firmware from current/next SONiC image. Both modular and non modular chassis platforms are supported.

FW update requires `platform_components.json` to be created and placed at: `sonic-buildimage/device/<platform_name>/<onie_platform>/platform_components.json`

Example:

1. Non modular chassis platform

```
{ "chassis":
  { "Chassis1":
    { "component":
      { "BIOS":
        {
          "firmware": "/usr/local/lib/firmware/<platform_name>/<onie_platform>/chassis1/bios.bin",
          "version": "<bios_version>"
        },
        "CPLD":
        {
          "firmware": "/usr/local/lib/firmware/<platform_name>/<onie_platform>/chassis1/cpld.bin",
          "version": "<cpld_version>"
        },
        "FPGA":
        {
          "firmware": "/usr/local/lib/firmware/<platform_name>/<onie_platform>/chassis1/fpga.bin",
          "version": "<fpga_version>"
        }
      }
    }
  }
}
```

2. Modular chassis platform



2. Modular chassis platform

```
{
  "chassis":
    { "Chassis1":
      { "component":
        {
          "BIOS":
            { "firmware": "/usr/local/lib/firmware/<platform_name>/ <onie_platform>/chassis1/bios.bin",
              "version": "<bios_version>" },
          "CPLD":
            { "firmware": "/usr/local/lib/firmware/<platform_name>/ <onie_platform>/chassis1/cpld.bin",
              "version": "<cpld_version>" },
          "FPGA":
            { "firmware": "/usr/local/lib/firmware/<platform_name>/ <onie_platform>/chassis1/fpga.bin",
              "version": "<fpga_version>" }
        }
      },
    },
  "module":
    { "Module1":
      { "component":
        {
          "CPLD":
            { "firmware": "/usr/local/lib/firmware/<platform_name>/ <onie_platform>/module1/cpld.bin",
              "version": "<cpld_version>" },
          "FPGA":
            { "firmware": "/usr/local/lib/firmware/<platform_name>/ <onie_platform>/module1/fpga.bin",
              "version": "<fpga_version>" }
        }
      },
    },
  }
}
```

- Usage:

```
config platform firmware update chassis component <component_name> fw [-y|--yes] [-f|--force] [-i|--image]
config platform firmware update module <module_name> component <component_name> fw [-y|--yes] [-f|--force] [-i|--image]
```

- Options:

- -y|--yes: automatic yes to prompts. Assume “yes” as answer to all prompts and run non-interactively
- -f|--force: update FW regardless the current version
- -i|--image: update FW using current/next SONiC image

Valid values:

- current
- next

Default:

- current



- Example:

```
admin@sonic:~$ sudo config platform firmware update chassis component BIOS fw
Warning: Immediate cold reboot is required to complete BIOS firmware update.
New firmware will be installed, continue? [y/N]: y
```

```
admin@sonic:~$ sudo config platform firmware update module Module1 component BIOS fw
Warning: Immediate cold reboot is required to complete BIOS firmware update.
New firmware will be installed, continue? [y/N]: y
```

- Note:

- FW update will be disabled if component definition is not provided (e.g., 'BIOS': { })
- FW version will be read from image if version field is not provided
- current/next values for `-i|-image` are taken from sonic-installer list

```
admin@sonic:~$ sudo sonic-installer list
```

2.2.2 PortChannels

2.2.2.1 PortChannel Show commands

show interfaces portchannel

This command displays all the port channels that are configured in the device and its current status.

- Usage:

```
show interfaces portchannel
```

- Example:

```
admin@sonic:~$ show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available, S -
selected, D - deselected
```

No.	Team	Dev	Protocol	Ports
24	PortChannel24	LACP(A)(Up)	Ethernet28(S) Ethernet24(S)	
48	PortChannel48	LACP(A)(Up)	Ethernet52(S) Ethernet48(S)	
40	PortChannel40	LACP(A)(Up)	Ethernet44(S) Ethernet40(S)	
0	PortChannel0	LACP(A)(Up)	Ethernet0(S) Ethernet4(S)	
8	PortChannel8	LACP(A)(Up)	Ethernet8(S) Ethernet12(S)	

2.2.2.2 PortChannel Config commands

This sub-section explains how to configure the portchannel and its member ports.

config portchannel

This command is used to add or delete the portchannel. It is recommended to use portchannel names in the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "PortChannel0002".

NOTE: If users specify any other name like "pc99", command will succeed, but such names are not supported. Such names are not printed properly in the "show interface portchannel" command. It is recommended not to use such names.

When any port is already member of any other portchannel and if user tries to add the same port in some other portchannel (without deleting it from the current portchannel), the command fails internally. But, it does not print any error message. In such cases, remove the member from current portchannel and then add it to new portchannel.



Command takes two optional arguments given below.

1. min-links - minimum number of links required to bring up the portchannel
2. fallback - true/false. LACP fallback feature can be enabled / disabled. When it is set to true, only one member port will be selected as active per portchannel during fallback mode. Refer https://github.com/sonic-net/SONiC/blob/master/doc/lag/LACP%20Fallback%20Feature%20for%20SONiC_v0.5.md for more details about fallback feature.
3. fast-rate - true/false, default is false (slow). Option specifying the rate in which we'll ask our link partner to transmit LACPDU packets in 802.3ad mode. **slow** - request partner to transmit LACPDUs every 30 seconds, **fast** - request partner to transmit LACPDUs every 1 second. In slow mode 60-90 seconds needed to detect linkdown, in fast mode only 2-3 seconds.

A port channel can be deleted only if it does not have any members or the members are already deleted. When a user tries to delete a port channel and the port channel still has one or more members that exist, the deletion of port channel is blocked.

- Usage:

```
config portchannel (add | del) <portchannel_name> [--min-links <num_min_links>] [--fallback (true | false)
[--fast-rate (true | false)]
```

- Example (Create the portchannel with name "PortChannel0011"):

```
admin@sonic:~$ sudo config portchannel add PortChannel0011
```

config portchannel member

This command adds or deletes a member port to/from the already created portchannel.

- Usage:

```
config portchannel member (add | del) <portchannel_name> <member_portname>
```

- Example (Add interface Ethernet4 as member of the portchannel "PortChannel0011"):

```
admin@sonic:~$ sudo config portchannel member add PortChannel0011 Ethernet4
```

Go Back To Beginning of the document or Beginning of this section

2.2.3 NVGRE

This section explains the various show commands and configuration commands available for users.

2.2.3.1 NVGRE show commands

This subsection explains how to display the NVGRE configuration.

show nvgre-tunnel

This command displays the NVGRE tunnel configuration.

- Usage:

```
show nvgre-tunnel
```

- Example:

```
admin@sonic:~$ show nvgre-tunnel
TUNNEL NAME    SRC IP
-----
tunnel_1       10.0.0.1
```






show nvgre-tunnel-map

This command displays the NVGRE tunnel map configuration.

- Usage:

```
show nvgre-tunnel-map
```

- Example:

```
admin@sonic:~$ show nvgre-tunnel-map
TUNNEL NAME      TUNNEL MAP NAME  VLAN ID  VSID
-----
tunnel_1         Vlan1000         1000     5000
tunnel_1         Vlan2000         2000     6000
```

2.2.3.2 NVGRE config commands

This subsection explains how to configure the NVGRE.

config nvgre-tunnel

This command is used to manage the NVGRE tunnel objects. It supports add/delete operations.

- Usage:

```
config nvgre-tunnel add <tunnel-name> --src-ip <source ip address> config nvgre-tunnel
delete <tunnel-name>
```

- Parameters:

- *tunnel-name*: the name of the NVGRE tunnel
- *src-ip*: source ip address

- Examples:

```
config nvgre-tunnel add 'tunnel_1' --src-ip '10.0.0.1' config nvgre-tunnel
delete 'tunnel_1'
```

config nvgre-tunnel-map

This command is used to manage the NVGRE tunnel map objects. It supports add/delete operations.

- Usage:

```
config nvgre-tunnel-map add <tunnel-name> <tunnel-map-name> --vlan-id <vlan> --vsid <virtual subnet>
config nvgre-tunnel-map delete <tunnel-name> <tunnel-map-name>
```

- Parameters:

- *tunnel-name*: the name of the NVGRE tunnel
- *tunnel-map-name*: the name of the NVGRE tunnel map
- *vlan-id*: VLAN identifier
- *vsid*: Virtual Subnet Identifier

- Examples:

```
config nvgre-tunnel-map add 'tunnel_1' 'Vlan2000' --vlan-id '2000' --vsid '6000' config nvgre-tunnel-map
delete 'tunnel_1' 'Vlan2000'
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.2.4 PBH

This section explains the various show commands and configuration commands available for users.

2.2.4.1 PBH show commands

This subsection explains how to display PBH configuration and statistics.

show pbh table

This command displays PBH table configuration.

- Usage:

```
show pbh table
```

- Example:

```
admin@sonic:~$ show pbh table
NAME          INTERFACE          DESCRIPTION
-----
pbh_table     Ethernet0          NVGRE and VxLAN
               Ethernet4
               PortChannel0001
               PortChannel0002
```

show pbh rule

This command displays PBH rule configuration.

- Usage:

```
show pbh rule
```

- Example:

```
admin@sonic:~$ show pbh rule
TABLE  RULE  PRIORITY  MATCH                                     HASH          ACTION          COUNTER
-----
Pbh_table  nvgre  2          ether_type: 0x0800                       inner_v6_hash  SET_ECMP_HASH  DISABLED
           ip_protocol: 0x2f
           gre_key: 0x2500/0xffffffff00
           inner_ether_type: 0x86dd
Pbh_table  vxlan  1          ether_type: 0x0800                       inner_v4_hash  SET_LAG_HASH   ENABLEED
           ip_protocol: 0x11
           l4_dst_port: 0x12b5
           inner_ether_type: 0x800
```



show pbh hash

This command displays PBH hash configuration.

- Usage:

```
show pbh hash
```

- Example:

```
admin@sonic:~$ show pbh hash
NAME                HASH FIELD
-----
inner_v4_hash        inner_ip_proto
                     inner_l4_dst_port
                     inner_l4_src_port
                     inner_dst_ipv4
                     inner_src_ipv4
inner_v6_hash        inner_ip_proto
                     inner_l4_dst_port
                     inner_l4_src_port
                     inner_dst_ipv6
                     inner_src_ipv6
```

show pbh hash-field

This command displays PBH hash field configuration.

- Usage:

```
show pbh hash-field
```

- Example:

```
admin@sonic:~$ show pbh hash-field
NAME                FIELD                MASK        SEQUENCE    SYMMETRIC
-----
inner_ip_proto       INNER_IP_PROTOCOL    N/A         2           Yes
inner_l4_dst_port    INNER_L4_DST_PORT    N/A         2           Yes
inner_l4_src_port    INNER_L4_SRC_PORT    255.0.0.0   3           Yes
inner_dst_ipv4       INNER_DST_IPV4       0.0.0.255   3           Yes
inner_src_ipv4       INNER_SRC_IPV4       ffff::      4           Yes
inner_dst_ipv6       INNER_DST_IPV6       ::ffff      4           Yes
inner_src_ipv6       INNER_SRC_IPV6
```

- Note:

- *SYMMETRIC* is an artificial column and is only used to indicate fields symmetry



show pbh statistics

This command displays PBH statistics.

- Usage:

```
show pbh statistics
```

- Example:

```
admin@sonic:~$ show pbh statistics
TABLE          RULE    RX PACKETS COUNT    RX BYTES COUNT
-----
pbh_table      nvgre    0                    0
pbh_table      vxlan    0                    0
```

- Note:

- *RX PACKETS COUNT* and *RX BYTES COUNT* can be cleared by user:

```
admin@sonic:~$ sonic-clear pbh statistics
```

2.2.4.2 PBH config commands

This subsection explains how to configure PBH.

config pbh table

This command is used to manage PBH table objects. It supports add/update/remove operations.

- Usage:

```
config pbh table add <table_name> --interface-list <interface_list> --description <description>
config pbh table update <table_name> [ --interface-list <interface_list> ] [ --description <description> ]
config pbh table delete <table_name>
```

- Parameters:

- *table_name*: the name of the PBH table
- *interface_list*: interfaces to which PBH table is applied
- *description*: the description of the PBH table

- Examples:

```
config pbh table add 'pbh_table' \
--interface-list 'Ethernet0,Ethernet4,PortChannel0001,PortChannel0002' \
--description 'NVGRE and VxLAN'
config pbh table update 'pbh_table' \
--interface-list 'Ethernet0'
config pbh table delete 'pbh_table'
```

config pbh rule

This command is used to manage PBH rule objects. It supports add/update/remove operations.

- Usage:



```

config pbh rule add <table_name> <rule_name> --priority <priority> \ [ --gre-key <gre_key> ]
[ --ether-type <ether_type> ] [ --ip-protocol <ip_protocol> ] \
[ --ipv6-next-header <ipv6_next_header> ] [ --l4-dst-port <l4_dst_port> ] [ --inner-ether-type <inner_ether_type> ] \
--hash <hash> [ --packet-action <packet_action> ] [ --flow-counter <flow_counter> ]
config pbh rule update <table_name> <rule_name> [ --priority <priority> ] \ [ --gre-key <gre_key> ] [ --ether-type
<ether_type> ] [ --ip-protocol <ip_protocol> ] \
[ --ipv6-next-header <ipv6_next_header> ] [ --l4-dst-port <l4_dst_port> ] [ --inner-ether-type <inner_ether_type> ] \
[ --hash <hash> ] [ --packet-action <packet_action> ] [ --flow-counter <flow_counter> ]
config pbh rule delete <table_name> <rule_name>

```

- Parameters:

- **table_name**: the name of the PBH table
- **rule_name**: the name of the PBH rule
- **priority**: the priority of the PBH rule
- **gre_key**: packet match for the PBH rule: GRE key (value/mask)
- **ether_type**: packet match for the PBH rule: EtherType (IANA Ethertypes)
- **ip_protocol**: packet match for the PBH rule: IP protocol (IANA Protocol Numbers)
- **ipv6_next_header**: packet match for the PBH rule: IPv6 Next header (IANA Protocol Numbers)
- **l4_dst_port**: packet match for the PBH rule: L4 destination port
- **inner_ether_type**: packet match for the PBH rule: inner EtherType (IANA Ethertypes)
- **hash**: hash object to apply with the PBH rule
- **packet_action**: packet action for the PBH rule

Valid values:

- SET_ECMP_HASH
- SET_LAG_HASH

Default:

- SET_ECMP_HASH

- **flow_counter**: packet/byte counter for the PBH rule

Valid values:

- DISABLED
- ENABLED

Default:

- DISABLED

- Examples:

```

config pbh rule add 'pbh_table' 'nvgre' \
--priority '2' \
--ether-type '0x0800' \
--ip-protocol '0x2f' \
--gre-key '0x2500/0xffffffff' \
--inner-ether-type '0x86dd' \
--hash 'inner_v6_hash' \
--packet-action 'SET_ECMP_HASH' \
--flow-counter 'DISABLED'
config pbh rule update 'pbh_table' 'nvgre' \
--flow-counter 'ENABLED'
config pbh rule delete 'pbh_table' 'nvgre'

```



config pbh hash

This command is used to manage PBH hash objects. It supports add/update/remove operations.

- Usage:

```
config pbh hash add <hash_name> --hash-field-list <hash_field_list>
config pbh hash update <hash_name> [ --hash-field-list <hash_field_list> ] config pbh hash delete
<hash_name>
```

- Parameters:

- *hash_name*: the name of the PBH hash
- *hash_field_list*: list of *hash-field* objects to apply with the PBH hash

- Examples:

```
config pbh hash add 'inner_v6_hash' \
--hash-field-list
'inner_ip_proto,inner_l4_dst_port,inner_l4_src_port,inner_dst_ipv6,inner_src_ipv6'
config pbh hash update 'inner_v6_hash' \
--hash-field-list 'inner_ip_proto'
config pbh hash delete 'inner_v6_hash'
```

config pbh hash-field

This command is used to manage PBH hash field objects. It supports add/update/remove operations.

- Usage

```
config pbh hash-field add <hash_field_name> \
--hash-field <hash_field> [ --ip-mask <ip_mask> ] --sequence-id <sequence_id>
config pbh hash-field update <hash_field_name> \
[ --hash-field <hash_field> ] [ --ip-mask <ip_mask> ] [ --sequence-id <sequence_id> ]
config pbh hash-field delete <hash_field_name>
```

- Parameters:

- *hash_field_name*: the name of the PBH hash field
- *hash_field*: native hash field for the PBH hash field
Valid values:
 - INNER_IP_PROTOCOL
 - INNER_L4_DST_PORT
 - INNER_L4_SRC_PORT
 - INNER_DST_IPV4
 - INNER_SRC_IPV4
 - INNER_DST_IPV6
 - INNER_SRC_IPV6
- *ip_mask*: IPv4/IPv6 address mask for the PBH hash field
Valid only: *hash_field* is:
 - INNER_DST_IPV4
 - INNER_SRC_IPV4
 - INNER_DST_IPV6
 - INNER_SRC_IPV6
- *sequence_id*: the order in which fields are hashed

- Examples:

```
config pbh hash-field add 'inner_dst_ipv6' \
--hash-field 'INNER_DST_IPV6' \
--ip-mask 'ffff:.' \
--sequence-id '4'
config pbh hash-field update 'inner_dst_ipv6' \
--ip-mask 'ffff:ffff:.'
config pbh hash-field delete 'inner_dst_ipv6'
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)





2.2.5 PINS

2.2.5.1 PINS Show commands

2.2.5.1.1 P4RT Table

show p4-table

This command displays the P4RT (P4 Runtime) tables in the application database.

These tables are used by PINS (P4 Integrated Network Stack) for orchagent to communicate with the P4RT application.

- Usage:

```
show p4-table
```

```
show p4-table <table_prefix>
```

- Example:

```
admin@sonic:~$ show p4-table
```

```
{
"P4RT_TABLE:ACL_TABLE_DEFINITION_TABLE:ACL_ACL_PRE_INGRESS_TABLE":
{
  "stage":"PRE_INGRESS",
  "match/dst_ipv6":
  {
    \ "bitwidth\ ":128,\ "format\ ":"IPV6\ ",
    \ "kind\ ":"sai_field\ ",\ "sai_field\ ":"SAI_ACL_TABLE_ATTR_FIELD_DST_IPV6\ "},
  "match/in_port":
  {
    \ "format\ ":"STRING\ ",
    \ "kind\ ":"sai_field\ ",
    \ "sai_field\ ":"SAI_ACL_TABLE_ATTR_FIELD_IN_PORT\ "},
  "match/is_ipv4":
  {
    \ "bitwidth\ ":1,
    \ "format\ ":"HEX_STRING\ ",
    \ "kind\ ":"sai_field\ ",
    \ "sai_field\ ":"SAI_ACL_TABLE_ATTR_FIELD_ACL_IP_TYPE/IPV4ANY\ "},
  "action/set_vrf": "[
  {
    \ "action\ ":"SAI_PACKET_ACTION_FORWARD\ "},
  {
    \ "action\ ":"SAI_ACL_ENTRY_ATTR_ACTION_SET_VRF\ ",
    \ "param\ ":"vrf_id\ "}]
  },
"P4RT_TABLE:ACL_ACL_PRE_INGRESS_TABLE":
{
  \ "match/dst_ip\ ":"10.53.192.0&255.255.240.0\ ",
  \ "match/is_ipv4\ ":"0x1\ ",
  \ "priority\ ":1132}:
  {
    "action": "set_vrf",
    "param/vrf_id": "p4rt-vrf-80",
    "controller_metadata": "my metadata" },
  ... }
```



The command supports filtering entries by table name. If a prefix is specified, only p4 table entries matching that prefix will be displayed

```
admin@sonic:~$ show p4-table ACL_ACL_PRE_INGRESS_TABLE
{
  "P4RT_TABLE:ACL_ACL_PRE_INGRESS_TABLE":
  {
    \"match/dst_ip\": \"10.53.192.0&255.255.240.0\",
    \"match/is_ipv4\": \"0x1\",
    \"priority\": 1132}:
    {
      \"action\": \"set_vrf\",
      \"param/vrf_id\": \"p4rt-vrf-80\",
      \"controller_metadata\": \"my metadata\"
    },
    ... }

```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.2.6 QoS

2.2.6.1 QoS Show commands

2.2.6.1.1 PFC

show pfc counters

This command displays the details of Rx & Tx priority-flow-control (pfc) for all ports. This command can be used to clear the counters using -c option.

- Usage:

```
show pfc counters
```

- Example:

```
admin@sonic:~$ show pfc counters
```

Port Rx	PFC0	PFC1	PFC2	PFC3	PFC4	PFC5	PFC6	PFC7
-----	-----	-----	-----	-----	-----	-----	-----	-----
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

Port Tx	PFC0	PFC1	PFC2	PFC3	PFC4	PFC5	PFC6	PFC7
-----	-----	-----	-----	-----	-----	-----	-----	-----
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

...

- NOTE: PFC counters can be cleared by the user with the following command:

```
admin@sonic:~$ sonic-clear pfccounters
```



show pfc asymmetric

This command displays the status of asymmetric PFC for all interfaces or a given interface.

- Usage:

```
show pfc asymmetric [<interface>]
```

- Example:

```
admin@sonic:~$ show pfc asymmetric
```

Interface	Asymmetric
-----	-----
Ethernet0	off
Ethernet2	off
Ethernet4	off
Ethernet6	off
Ethernet8	off
Ethernet10	off
Ethernet12	off
Ethernet14	off

```
admin@sonic:~$ show pfc asymmetric Ethernet0
```

Interface	Asymmetric
-----	-----
Ethernet0	off

show pfc priority

This command displays the lossless priorities for all interfaces or a given interface.

- Usage:

```
show pfc priority [<interface>]
```

- Example:

```
admin@sonic:~$ show pfc priority
```

Interface	Lossless priorities
-----	-----
Ethernet0	3,4
Ethernet2	3,4
Ethernet8	3,4
Ethernet10	3,4
Ethernet16	3,4

```
admin@sonic:~$ show pfc priority Ethernet0
```

Interface	Lossless priorities
-----	-----
Ethernet0	3,4



2.2.6.1.2 Queue And Priority-Group

This sub-section explains the following queue parameters that can be displayed using “show queue” command.

1. queue counters
2. queue watermark
3. priority-group watermark
4. queue persistent-watermark
5. queue wredcounters

show queue counters

This command displays packet and byte counters for all queues of all ports or one specific-port given as argument. This command can be used to clear the counters for all queues of all ports. Note that port specific clear is not supported.

- Usage:

```
show queue counters [<interface_name>]
```

- Example:

```
admin@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	UC8	0	0	0	0
Ethernet0	UC9	0	0	0	0
Ethernet0	MC0	0	0	0	0
Ethernet0	MC1	0	0	0	0
Ethernet0	MC2	0	0	0	0
Ethernet0	MC3	0	0	0	0
Ethernet0	MC4	0	0	0	0
Ethernet0	MC5	0	0	0	0
Ethernet0	MC6	0	0	0	0
Ethernet0	MC7	0	0	0	0
Ethernet0	MC8	0	0	0	0
Ethernet0	MC9	0	0	0	0

Port	TxQ	Counter/pkts	Counter/bytes	Drop/pkts	Drop/bytes
Ethernet4	UC0	0	0	0	0
Ethernet4	UC1	0	0	0	0
Ethernet4	UC2	0	0	0	0
Ethernet4	UC3	0	0	0	0
Ethernet4	UC4	0	0	0	0
Ethernet4	UC5	0	0	0	0
Ethernet4	UC6	0	0	0	0
Ethernet4	UC7	0	0	0	0
Ethernet4	UC8	0	0	0	0
Ethernet4	UC9	0	0	0	0
Ethernet4	MC0	0	0	0	0
Ethernet4	MC1	0	0	0	0
Ethernet4	MC2	0	0	0	0
Ethernet4	MC3	0	0	0	0
Ethernet4	MC4	0	0	0	0
Ethernet4	MC5	0	0	0	0
Ethernet4	MC6	0	0	0	0
Ethernet4	MC7	0	0	0	0
Ethernet4	MC8	0	0	0	0
Ethernet4	MC9	0	0	0	0

...



Optionally, you can specify an interface name in order to display only that particular interface

- Example:

```
admin@sonic:~$ show queue counters Ethernet72
```

- NOTE: Queue counters can be cleared by the user with the following command:

```
admin@sonic:~$ sonic-clear queuecounters
```

show queue watermark

This command displays the user watermark for the queues (Egress shared pool occupancy per queue) for either the unicast queues or multicast queues for all ports

- Usage:

```
show queue watermark (multicast | unicast)
```

- Example:

```
admin@sonic:~$ show queue watermark unicast
Egress shared pool occupancy 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

admin@sonic:~$ show queue watermark multicast (Egress shared pool occupancy per
multicast queue)
```

show priority-group

This command displays:

1. The user watermark or persistent-watermark for the Ingress “headroom” or “shared pool occupancy” per priority-group for all ports.
2. Dropped packets per priority-group for all ports

- Usage:

```
show priority-group (watermark | persistent-watermark) (headroom | shared)
show priority-group drop counters
```

- Example:

```
admin@sonic:~$ show priority-group watermark shared
Ingress shared pool occupancy 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
```

Example (Ingress headroom per PG):

```
admin@sonic:~$ show priority-group watermark headroom
```

- Example (Ingress shared pool occupancy per PG):

```
admin@sonic:~$ show priority-group persistent-watermark shared
```

- Example (Ingress headroom per PG):

```
admin@sonic:~$ show priority-group persistent-watermark headroom
```



- Example (Ingress dropped packets per PG):

```
admin@sonic:~$ show priority-group drop counters Ingress PG
dropped packets:
```

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

In addition to user watermark(“show queue|priority-group watermark ...”), a persistent watermark is available. It hold values independently of user watermark. This way user can use “user watermark” for debugging, clear it, etc, but the “persistent watermark” will not be affected.

show queue persistent-watermark

This command displays the user persistet-watermark for the queues (Egress shared pool occupancy per queue) for either the unicast queues or multicast queues for all ports

- Usage:

```
show queue persistent-watermark (unicast | multicast)
```

- Example:

```
admin@sonic:~$ show queue persistent-watermark unicast
Egress shared pool occupancy per unicast queue:
```

Port	UC0	UC1	UC2	UC3	UC4	UC5	UC6	UC7
-----	----	----	----	----	----	----	----	----
Ethernet0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ethernet4	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ethernet8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ethernet12	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

- Example (Egress shared pool occupancy per multicast queue):

```
admin@sonic:~$ show queue persistent-watermark multicast
```

- NOTE: “user watermark”, “persistent watermark” and “ingress dropped packets” can be cleared by user:

```
admin@sonic:~$ sonic-clear queue persistent-watermark unicast
admin@sonic:~$ sonic-clear queue persistent-watermark multicast
admin@sonic:~$ sonic-clear priority-group persistent-watermark shared
admin@sonic:~$ sonic-clear priority-group persistent-watermark headroom
admin@sonic:~$ sonic-clear priority-group drop counters
```

show queue wredcounters

This command displays wred-drop packet/byte and ecn-marked packet/byte counters for all queues of all ports or one specific-port given as argument. This command can be used to clear the counters for all queues of all ports. Note that port specific clear is not supported.

- Usage:

```
show queue wredcounters [<interface_name>]
```



Example:

```
admin@sonic:~$ show queue wredcounters
```

Port	TxQ	WredDrp/pkts	WredDrp/bytes	EcnMarked/pkts	EcnMarked/ 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	UC8	0	0	0	0
Ethernet0	UC9	0	0	0	0
Ethernet0	MC0	0	0	0	0
Ethernet0	MC1	0	0	0	0
Ethernet0	MC2	0	0	0	0
Ethernet0	MC3	0	0	0	0
Ethernet0	MC4	0	0	0	0
Ethernet0	MC5	0	0	0	0
Ethernet0	MC6	0	0	0	0
Ethernet0	MC7	0	0	0	0
Ethernet0	MC8	0	0	0	0
Ethernet0	MC9	0	0	0	0
...					

Optionally, you can specify an interface name in order to display only that particular interface

- Example:

```
admin@sonic:~$ show queue wredcounters Ethernet72
```

- NOTE: Queue counters can be cleared by the user with the following command:

```
admin@sonic:~$ sonic-clear queue wredcounters
```

2.2.6.1.3 Buffer Pool

This sub-section explains the following buffer pool parameters that can be displayed using “show buffer_pool” command.

1. buffer pool watermark
2. buffer pool persistent-watermark

show buffer_pool watermark

This command displays the user watermark for all the buffer pools

- Usage:

```
show buffer_pool watermark
```

- Example:

```
admin@sonic:~$ show buffer_pool watermark Shared pool maximum occupancy:
```

Pool	Bytes
-----	-----
ingress_lossless_pool	0
lossy pool	2464





2.2.6.2 QoS config commands

config qos clear

This command is used to clear all the QoS configuration from all the following QoS Tables in ConfigDB.

1. TC_TO_PRIORITY_GROUP_MAP,
2. MAP_PFC_PRIORITY_TO_QUEUE,
3. TC_TO_QUEUE_MAP,
4. DSCP_TO_TC_MAP,
5. MPLS_TC_TO_TC_MAP,
6. SCHEDULER,
7. PFC_PRIORITY_TO_PRIORITY_GROUP_MAP,
8. PORT_QOS_MAP,
9. WRED_PROFILE,
10. QUEUE,
11. CABLE_LENGTH,
12. BUFFER_POOL,
13. BUFFER_PROFILE,
14. BUFFER_PG,
15. BUFFER_QUEUE

- Usage:

```
config qos clear
```

- Example:

```
admin@sonic:~$ sudo config qos clear
```

config qos reload

This command is used to reload the QoS configuration. QoS configuration has got two sets of configurations.

1. Generic QOS Configuration - This gives complete list of all possible QOS configuration. Its given in the file /usr/share/sonic/templates/qos_config.j2 in the device. Reference: https://github.com/sonic-net/sonic-buildimage/blob/master/files/build_templates/qos_config.j2 Users have flexibility to have platform specific qos configuration by placing the qos_config.j2 file at /usr/share/sonic/device/. If users want to modify any of this loaded QOS configuration, they can modify this file in the device and then issue the “config qos reload” command.
2. Platform specific buffer configuration. Every platform has got platform specific and topology specific (T0 or T1 or T2) buffer configuration at /usr/share/sonic/device//buffers_defaults_tx.j2 In addition to platform specific configuration file, a generic configuration file is also present at /usr/share/sonic/templates/buffers_config.j2. Reference: https://github.com/sonic-net/sonic-buildimage/blob/master/files/build_templates/buffers_config.j2 Users can either modify the platform specific configuration file, or the generic configuration file and then issue this “config qos reload” command.



These configuration files are already loaded in the device as part of the reboot process. In case if users wants to modify any of these configurations, they need to modify the appropriate QoS tables and fields in these files and then use this reload command. This command uses those modified buffers.json.j2 file & qos.json.j2 file and reloads the new QoS configuration. If users have not made any changes in these configuration files, this command need not be executed.

Some of the example QoS configurations that users can modify are given below.

1. TC_TO_PRIORITY_GROUP_MAP
2. MAP_PFC_PRIORITY_TO_QUEUE
3. TC_TO_QUEUE_MAP
4. DSCP_TO_TC_MAP
5. MPLS_TC_TO_TC_MAP
6. SCHEDULER
7. PFC_PRIORITY_TO_PRIORITY_GROUP_MAP
8. PORT_QOS_MAP
9. WRED_PROFILE
10. CABLE_LENGTH
11. BUFFER_QUEUE

- Usage:

```
config qos reload
```

- Example:

```
admin@sonic:~$ sudo config qos reload
```

In this example, it uses the buffers.json.j2 file and qos.json.j2 file from platform specific folders.

When there are no changes in the platform specific configuration files, they internally use the file "/usr/share/sonic/templates/buffers_config.j2" and "/usr/share/sonic/templates/qos_config.j2" to generate the configuration.

When an error occurs, such as "Operation not completed successfully, please save and reload configuration," the system will record the status, after executing all the latter commands, exit with code 1.

config qos reload -ports port_list

This command is used to reload the default QoS configuration on a group of ports. Typically, the default QoS configuration is in the following tables.

1. PORT_QOS_MAP
2. QUEUE
3. BUFFER_PG
4. BUFFER_QUEUE
5. BUFFER_PORT_INGRESS_PROFILE_LIST
6. BUFFER_PORT_EGRESS_PROFILE_LIST
7. CABLE_LENGTH



If there was QoS configuration in the above tables for the ports:

- if `--force` option is provided, the existing QoS configuration will be replaced by the default QoS configuration,
- otherwise, the command will exit with nothing updated.

- Usage:

```
config qos reload --ports <port>[,port]
```

- Example:

```
admin@sonic:~$ sudo config qos reload --ports Ethernet0,Ethernet4
```

In this example, it updates the QoS configuration on port Ethernet0 and Ethernet4 to default.

If there was QoS configuration on the ports, the command will clear the existing QoS configuration on the port and reload to default.

[Go Back To Beginning of the document](#) or [Beginning of this section](#)

2.2.7 Radius

2.2.7.1 show radius commands

This command displays the global radius configuration that includes the auth_type, retransmit, timeout and passkey.

- Usage:

```
show radius
```

- Example:

```
admin@sonic:~$ show radius
```

```
RADIUS global auth_type pap (default) RADIUS global retransmit 3 (default) RADIUS global timeout 5 (default)
```

```
RADIUS global passkey <EMPTY_STRING> (default)
```

2.2.7.2 Radius config commands

This command is to config the radius server for various parameter listed.

- Usage:

```
config radius
```

- Example:

```
admin@sonic:~$ config radius
```

```
add authtype Specify a RADIUS server
```

```
default delete Specify RADIUS server global auth_type [chap | pap | mschapv2] set its default
```

```
nasip passkey configuration
```

```
retransmit Delete a RADIUS server
```

```
sourceip Specify RADIUS server global NAS-IP|IPv6-Address <IPAddress> Specify RADIUS
```

```
statistics server global passkey <STRING>
```

```
timeout Specify RADIUS server global retry attempts <0 - 10> Specify RADIUS
```

```
server global source ip <IPAddress>
```

```
Specify RADIUS server global statistics [enable | disable | ...]
```

```
Specify RADIUS server global timeout <1 - 60>
```



2.2.8 sFlow

2.2.8.1 sFlow Show commands

show sflow

This command displays the global sFlow configuration that includes the admin state, collectors, the Agent ID and counter polling interval.

- Usage:

```
show sflow
```

- Example:

```
admin@sonic:~# show sflow
sFlow Global Information:
sFlow Admin State:      up
sFlow Sample Direction: both
sFlow Polling Interval: default
sFlow AgentID:          lo

2 Collectors configured:
  Name: collector_A      IP addr: 10.11.46.2  UDP port: 6343
  Name: collector_lo     IP addr: 127.0.0.1   UDP port: 6343
```

show sflow interface

This command displays the per-interface sflow admin status and the sampling rate.

- Usage:

```
show sflow interface
```

- Example:

```
admin@sonic:~# show sflow interface

sFlow interface configurations
+-----+-----+-----+-----+
| Interface | Admin State | Sampling Rate | Sampling Direction |
+-----+-----+-----+-----+
| Ethernet0 | up          | 4000          | both               |
+-----+-----+-----+-----+
| Ethernet1 | up          | 4000          | tx                 |
+-----+-----+-----+-----+
| ...      |             |               |                   |
+-----+-----+-----+-----+
| Ethernet61 | up          | 4000          | rx                 |
+-----+-----+-----+-----+
| Ethernet62 | up          | 4000          | tx                 |
+-----+-----+-----+-----+
| Ethernet63 | up          | 4000          | both               |
+-----+-----+-----+-----+
```



2.2.8.2 sFlow Config commands

config sflow collector add

This command is used to add a sFlow collector. Note that a maximum of 2 collectors is allowed.

- Usage:

```
config sflow collector add <collector-name> <ipv4-address | ipv6-address> [port <number>]
```

- Parameters:

- collector-name: unique name of the sFlow collector
- ipv4-address : IP address of the collector in dotted decimal format for IPv4
- ipv6-address : x: x: x: x: x: x: x: x format for IPv6 address of the collector (where :: notation specifies successive hexadecimal fields of zeros)
- port (OPTIONAL): specifies the UDP port of the collector (the range is from 0 to 65535. The default is 6343.)

- Example:

```
admin@sonic:~# sudo config sflow collector add collector_A 10.11.46.2
```

config sflow collector del

This command is used to delete a sFlow collector with the given name.

- Usage:

```
config sflow collector del <collector-name>
```

- Parameters:

- collector-name: unique name of the sFlow collector

- Example:

```
admin@sonic:~# sudo config sflow collector del collector_A
```

config sflow agent-id

This command is used to add/delete the sFlow agent-id. This setting is global (applicable to both collectors) and optional. Only a single agent-id is allowed. If agent-id is not specified (with this CLI), an appropriate IP that belongs to the switch is used as the agent-id based on some simple heuristics.

- Usage:

```
config sflow agent-id <add|del> <interface-name>
```

- Parameters:

- interface-name: specify the interface name whose ipv4 or ipv6 address will be used as the agent-id in sFlow datagrams.

- Example:

```
admin@sonic:~# sudo config sflow agent-id add lo
```

config sflow

Globally, sFlow is disabled by default. When sFlow is enabled globally, the sflow daemon is started and sampling will start on all interfaces which have sFlow enabled at the interface level (see “config sflow interface...”). When sflow is disabled globally, sampling is stopped on all relevant interfaces and sflow daemon is stopped.

- Usage:

```
config sflow <enable|disable>
```

- Example:

```
admin@sonic:~# sudo config sflow enable
```



config sflow sample-direction

This command takes global sflow sample direction. If not configured, default is “rx” for backward compatibility. Based on the direction, the sFlow is enabled at all the interface level at rx or tx or both.

- Usage:

```
config sflow sample-direction <rx|tx|both>
```

- Example:

```
admin@sonic:~# sudo config sflow sample-direction tx
```

config sflow interface

Enable/disable sflow at an interface level. By default, sflow is enabled on all interfaces at the interface level. Use this command to explicitly disable sFlow for a specific interface. An interface is sampled if sflow is enabled globally as well as at the interface level. Note that this configuration deals only with sFlow flow samples and not counter samples.

- Usage:

```
config sflow interface <enable|disable> <interface-name|all>
```

- Parameters:

- interface-name: specify the interface for which sFlow flow samples have to be enabled/disabled. The “all” keyword is used as a convenience to enable/disable sflow at the interface level for all the interfaces.

- Example:

```
admin@sonic:~# sudo config sflow interface disable Ethernet40
```

config sflow interface sample-direction

Set sample direction to determine ingress sampling or egress sampling or both. If not configured, default is “rx”.

- Usage:

```
config sflow sample-direction <interface-name|all> <rx|tx|both>
```

- Parameters:

- interface-name: specify the interface for which sFlow flow sample-direction has to be set. The “all” keyword is used as a convenience to set sflow sample-direction at the interface level for all the interfaces.

- Example:

```
admin@sonic:~# sudo config sflow interface sample-direction Ethernet40 tx
```

Note: The local configuration applied to an interface has higher precedence over the global configuration provided through the “all” keyword.



config sflow interface sample-rate

Configure the sample-rate for a specific interface.

The default sample rate for any interface is (ifSpeed / 1e6) where ifSpeed is in bits/sec. So, the default sample rate based on interface speed is:

1-in-1000 for a 1G link
 1-in-10,000 for a 10G link 1-in-
 40,000 for a 40G link 1-in-50,000 for
 a 50G link
 1-in-100,000 for a 100G link

It is recommended not to change the defaults. This CLI is to be used only in case of exceptions (e.g., to set the sample-rate to the nearest power-of-2 if there are hardware restrictions in using the defaults)

- Usage:

```
config sflow interface sample-rate <interface-name> <value>
```

- Parameters:

- interface-name: specify the interface for which the sampling rate value is to be set
- value: value is the average number of packets skipped before the sample is taken. “The sampling rate specifies random sampling probability as the ratio of packets observed to samples generated. For example a sampling rate of 256 specifies that, on average, 1 sample will be generated for every 256 packets observed.” Valid range 256:8388608.

- Example:

```
admin@sonic:~# sudo config sflow interface sample-rate Ethernet32 1000
```

config sflow polling-interval

This command is used to set the counter polling interval. Default is 20 seconds.

- Usage:

```
config sflow polling-interval <value>
```

- Parameters:

- value: 0-300 seconds. Set polling-interval to 0 to disable counter polling

- Example:

```
admin@sonic:~# sudo config sflow polling-interval 30
```

Go Back To Beginning of the document or Beginning of this section



2.2.9 SNMP

2.2.9.1 SNMP Show commands

show runningconfiguration snmp

This command displays the global SNMP configuration that includes the location, contact, community, and user settings.

- Usage:

```
show runningconfiguration snmp
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp Location
```

show runningconfiguration snmp location

This command displays the SNMP location setting.

- Usage:

```
show runningconfiguration snmp location
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp location Location
```

- Usage:

```
show runningconfiguration snmp location --json
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp location --json {'Location': 'Your City'}
```

show runningconfiguration snmp contact

This command displays the SNMP contact setting.

- Usage:

```
show runningconfiguration snmp contact
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp contact
```

- Usage:

```
show runningconfiguration snmp contact --json
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp contact --json {'xxx':  
'xxx@yourdomain.xxx'}
```

show runningconfiguration snmp community

This command display the SNMP community settings.

- Usage:

```
show runningconfiguration snmp community
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp community
```

- Usage:

```
show runningconfiguration snmp community --json
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp community --json  
{  
  'xxx':  
    {'TYPE': 'RW'}  
}
```




show runningconfiguration snmp user

This command display the SNMP user settings.

- Usage:

```
show runningconfiguration snmp user
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp user
```

User	Permission	Type	Auth Type	Auth Password	Encryption Type	Encryption Password
-----	-----	----	-----	-----	-----	-----
Yourname	RO	Priv	SHA	YourAuthPass	AES	YourPassword

- Usage:

```
show runningconfiguration snmp user --json
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp user --json
```

```
{ 'YourName':
  {
    'SNMP_USER_TYPE': 'Priv',
    'SNMP_USER_PERMISSION': 'RO',
    'SNMP_USER_AUTH_TYPE': 'SHA',
    'SNMP_USER_AUTH_PASSWORD': 'YourAuthPass',
    'SNMP_USER_ENCRYPTION_TYPE': 'AES',
    'SNMP_USER_ENCRYPTION_PASSWORD': 'YourEncryptPass'
  }
}
```



2.2.9.2 SNMP Config commands

This sub-section explains how to configure SNMP.

config snmp location add/del/modify

This command is used to add, delete, or modify the SNMP location.

- Usage:

```
config snmp location (add | del | modify) <location>
```

- Example (Add new SNMP location “Your Location” if it does not already exist):

```
admin@sonic:~$ sudo config snmp location add Your Location
SNMP Location Your Location has been added to configuration Restarting SNMP service...
```

- Example (Delete SNMP location “Your Location” if it already exists):

```
admin@sonic:~$ sudo config snmp location del Your Location
SNMP Location Your Location removed from configuration Restarting SNMP service...
```

- Example (Modify SNMP location “Your Location” to “OtherLocation”):

```
admin@sonic:~$ sudo config snmp location modify OtherLocation
SNMP location OtherLocation modified in configuration Restarting SNMP service...
```

config snmp contact add/del/modify

This command is used to add, delete, or modify the SNMP contact.

- Usage:

```
config snmp contact add <contact> <contact_email>
```

- Example:

```
admin@sonic:~$ sudo config snmp contact add YourName YourName@yourdomain.xxx
Contact name YourName and contact email YourName@yourdomain.xxx have been added to configuration
Restarting SNMP service...
```

- Usage:

```
config snmp contact del <contact>
```

- Example:

```
admin@sonic:~$ sudo config snmp contact del YourName SNMP contact YourName removed from configuration
Restarting SNMP service...
```

- Usage:

```
config snmp contact modify <contact> <contact_email>
```

- Example:

```
admin@sonic:~$ sudo config snmp contact modify OtherName OtherName@yourdomain.xxx
SNMP contact OtherName and contact email OtherName@yourdomain.com updated Restarting SNMP service...
```



config snmp community add/del/replace

This command is used to add, delete, or replace the SNMP community.

- Usage:

```
config snmp community add <community> (RO | RW)
```

- Example:

```
admin@sonic:~$ sudo config snmp community add YourCommunity ro
SNMP community YourCommunity added to configuration
Restarting SNMP service...
```

- Usage:

```
config snmp community del <community>
```

- Example:

```
admin@sonic:~$ sudo config snmp community del YourCommunity
SNMP community YourCommunity removed from configuration Restarting SNMP service...
```

- Usage:

```
config snmp community replace <community> <new_community>
```

- Example:

```
admin@sonic:~$ sudo config snmp community replace YourCommunity NewCommunity
SNMP community NewCommunity added to configuration
SNMP community NewCommunity replace community YourCommunity Restarting SNMP service...
```

config snmp user add/del

This command is used to add or delete the SNMP user for SNMPv3.

- Usage:

```
config snmp user add <user> (noAuthNoPriv | AuthNoPriv | Priv) (RO | RW) [[(MD5 | SHA | MMAC-SHA-2)
<auth_password>] [(DES |AES) <encrypt_password>]]
```

- Example:

```
admin@sonic:~$ sudo config snmp user add testuser1 noauthnopriv ro
SNMP user testuser1 added to configuration Restarting SNMP service...
```

- Example:

```
admin@sonic:~$ sudo config snmp user add testuser2 authnopriv ro sha testuser2_auth_pass
SNMP user testuser2 added to configuration Restarting SNMP service...
```

- Example:

```
admin@sonic:~$ sudo config snmp user add testuser3 priv rw md5 testuser3_auth_pass aes testuser3_encrypt_pass
SNMP user testuser3 added to configuration Restarting SNMP service...
```

- Usage:

```
config snmp user del <user>
```

- Example:

```
admin@sonic:~$ sudo config snmp user del testuser1
SNMP user testuser1 removed from configuration Restarting SNMP service...
```

Go Back To Beginning of the document or Beginning of this section



2.2.10 Startup & Running Configuration

2.2.10.1 Startup Configuration

show startupconfiguration bgp

This command is used to display the startup configuration for the BGP module.

- Usage:

```
show startupconfiguration bgp
```

- Example:

```
admin@sonic:~$ show startupconfiguration bgp Routing-
Stack is: quagga
!
! ===== Managed by sonic-cfggen DO NOT edit manually! =====
! generated by templates/quagga/bgpd.conf.j2 with config DB data
! file: bgpd.conf
!
!
hostname T1-2
password YourPassword
log syslog informational
log facility local4
! enable password !
!
! bgp multiple-instance
!
route-map FROM_BGP_SPEAKER_V4 permit 10
!
route-map TO_BGP_SPEAKER_V4 deny 10
!
router bgp 65000
  bgp log-neighbor-changes
  bgp bestpath as-path multipath-relax no bgp
  default ipv4-unicast
  bgp graceful-restart restart-time 180
<Only the partial output is shown here. In actual command, more configuration information will be
displayed>
```



2.2.10.2 Running Configuration

This sub-section explains the show commands for displaying the running configuration for the following modules.

1. bgp
2. interfaces
3. ntp
4. snmp
5. all
6. acl
7. ports
8. syslog

show runningconfiguration all

This command displays the entire running configuration.

- Usage:

```
show runningconfiguration all
```

- Example:

```
admin@sonic:~$ show runningconfiguration all
```

show runningconfiguration bgp

This command displays the running configuration of the BGP module.

- Usage:

```
show runningconfiguration bgp
```

- Example:

```
admin@sonic:~$ show runningconfiguration bgp
```

show runningconfiguration interfaces

This command displays the running configuration for the “interfaces”.

- Usage:

```
show runningconfiguration interfaces
```

- Example:

```
admin@sonic:~$ show runningconfiguration interfaces
```

show runningconfiguration ntp

This command displays the running configuration of the ntp module.

- Usage:

```
show runningconfiguration ntp
```

- Example:

```
admin@sonic:~$ show runningconfiguration ntp NTP Servers
-----
1.1.1.1 2.2.2.2
```



show runningconfiguration syslog

This command displays the running configuration of the syslog module.

- Usage:

```
show runningconfiguration syslog
```

- Example:

```
admin@sonic:~$ show runningconfiguration syslog Syslog
Servers
-----
4.4.4.4
5.5.5.5
```

show runningconfiguration snmp

This command displays the running configuration of the snmp module.

- Usage:

```
show runningconfiguration snmp
```

- Example:

```
admin@sonic:~$ show runningconfiguration snmp
```

show runningconfiguration acl

This command displays the running configuration of the acls

- Usage:

```
show runningconfiguration acl
```

- Example:

```
admin@sonic:~$ show runningconfiguration acl
```

show runningconfiguration ports

This command displays the running configuration of the ports

- Usage:

```
show runningconfiguration ports [<portname>]
```

- Examples:

```
admin@sonic:~$ show runningconfiguration ports
admin@sonic:~$ show runningconfiguration ports Ethernet0
```

Go Back To Beginning of the document or Beginning of this section



2.2.11 Static routing

2.2.11.1 Static routing Config Commands

This sub-section explains of commands is used to add or remove the static route.

config route add

This command is used to add a static route. Note that prefix /nexthop vrf's and interface name are optional.

- Usage:

```
config route add prefix [vrf <vrf>] <A.B.C.D/M> nexthop [vrf <vrf>] <A.B.C.D> dev <interface name>
```

- Example:

```
admin@sonic:~$ config route add prefix 2.2.3.4/32 nexthop 30.0.0.9
admin@sonic:~$ config route add prefix 4.0.0.0/24 nexthop dev Ethernet32.10
```

It also supports ECMP, and adding a new nexthop to the existing prefix will complement it and not overwrite them.

- Example:

```
admin@sonic:~$ sudo config route add prefix 2.2.3.4/32 nexthop vrf Vrf-RED 30.0.0.9
admin@sonic:~$ sudo config route add prefix 2.2.3.4/32 nexthop vrf Vrf-BLUE 30.0.0.10
```

config route del

This command is used to remove a static route. Note that prefix /nexthop vrf's and interface name are optional.

- Usage:

```
config route del prefix [vrf <vrf>] <A.B.C.D/M> nexthop [vrf <vrf>] <A.B.C.D> dev <interface name>
```

- Example:

```
admin@sonic:~$ sudo config route del prefix 2.2.3.4/32 nexthop vrf Vrf-RED 30.0.0.9
admin@sonic:~$ sudo config route del prefix 2.2.3.4/32 nexthop vrf Vrf-BLUE 30.0.0.10
```

This sub-section explains of command is used to show current routes.

show ip route

- Usage:

```
show ip route
```

- Example:

```
admin@sonic:~$ show ip route
Codes:
K - kernel route,      C - connected,    S - static,      R - RIP,        O - OSPF,      I - IS-IS,
B - BGP,              E - EIGRP,      N - NHRP,        T - Table,      v - VNC,       V - VNC-Direct,
A - Babel,            D - SHARP,      F - PBR,         f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
S>* 0.0.0.0/0 [200/0] via 192.168.111.3, eth0, weight 1, 3d03h58m
S>  1.2.3.4/32 [1/0] via 30.0.0.7, weight 1, 00:00:06
C>* 10.0.0.18/31 is directly connected, Ethernet36, 3d03h57m
C>* 10.0.0.20/31 is directly connected, Ethernet40, 3d03h57m
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.2.12 Subinterfaces

2.2.12.1 Subinterfaces Show Commands

show subinterfaces status

This command displays all the subinterfaces that are configured on the device and its current status.

- Usage:

```
show subinterfaces status
```

- Example:

```
admin@sonic:~$ show subinterfaces status
```

Sub port interface	Speed	MTU	Vlan	Admin	TYPE
-----	-----	-----	-----	-----	-----
Eth64.10	100G	9100	100	up	dot1q-encapsulation
Ethernet0.100	100G	9100	100	up	dot1q-encapsulation

2.2.12.2 Subinterfaces Config Commands

This sub-section explains how to configure subinterfaces.

config subinterface

- Usage:

```
config subinterface (add | del) <subinterface_name> [vlan <1-4094>]
```

- Example (Create the subinterfces with name “Ethernet0.100”):

```
admin@sonic:~$ sudo config subinterface add Ethernet0.100
```

- Example (Create the subinterfces with name “Eth64.100”):

```
admin@sonic:~$ sudo config subinterface add Eth64.100 100
```

- Example (Delete the subinterfces with name “Ethernet0.100”):

```
admin@sonic:~$ sudo config subinterface del Ethernet0.100
```

- Example (Delete the subinterfces with name “Eth64.100”):

```
admin@sonic:~$ sudo config subinterface del Eth64.100 100
```

Go Back To Beginning of the document or Beginning of this section

2.2.13 Switchport Modes

2.2.13.1 Switchport Modes Config Commands

This subsection explains how to configure switchport modes on a Port/PortChannel. ****config switchport mode**** Usage:

```
config switchport mode <access|trunk|routed> <member_portname/member_portchannel>
```

- Example (Config switchport mode access on “Ethernet0”):

```
admin@sonic:~$ sudo config switchport mode access Ethernet0
```

- Example (Config switchport mode trunk on “Ethernet4”):

```
admin@sonic:~$ sudo config switchport mode trunk Ethernet4
```

- Example (Config switchport mode routed on “Ethernet12”):

```
admin@sonic:~$ sudo config switchport mode routed Ethernet12
```

Go Back To Beginning of the document or Beginning of this section



2.14 Syslog

2.2.14.1 Syslog Show Commands

This subsection explains how to display syslog related configuration.

show syslog

This command displays configured syslog servers.

- Usage:

```
show syslog
```

- Example:

```
admin@sonic:~$ show syslog
SERVER IP      SOURCE IP      PORT    VRF
-----
2.2.2.2        1.1.1.1        514     default
```

show syslog rate-limit-host

This command displays rate limit configuration for host.

- Usage

```
show syslog rate-limit-host
```

- Example:

```
admin@sonic:~$ show syslog rate-limit-host
INTERVAL      BURST
-----
500           50000
```

show syslog rate-limit-container

This command displays rate limit configuration for containers.

- Usage

```
show syslog rate-limit-container [<service_name>] -n [<namespace>]
```

- Example:

```
admin@sonic:~$ show syslog rate-limit-container
SERVICE      INTERVAL      BURST
-----
bgp            0             0
database       300           20000
lldp           300           20000
mgmt.-framework 300           20000
pmon           300           20000
radv           300           20000
snmp           300           20000
swss           300           20000
syncd          300           20000
teamd          300           20000
telemetry      300           20000
```



```
admin@sonic:~$ show syslog rate-limit-
container bgp
```

SERVICE	INTERVAL	BURST
-----	-----	-----
bgp	0	0

```
# Multi ASIC
```

```
show syslog rate-limit-container
```

SERVICE -	INTERVAL	BURST
-----	-----	-----
bgp	500	N/A
snmp	300	20000
swss	2000	12000

```
Namespace asic0:
```

SERVICE -	INTERVAL	BURST
-----	-----	-----
bgp	500	N/A
snmp	300	20000
swss	2000	12000

```
# Multi ASIC
```

```
show syslog rate-limit-container bgp
```

SERVICE -	INTERVAL	BURST
-----	-----	-----
bgp	500	5000

```
Namespace asic0:
```

SERVICE -	INTERVAL	BURST
-----	-----	-----
bgp	500	5000

```
# Multi ASIC
```

```
show syslog rate-limit-container bgp -n asic1 Namespace
```

```
asic1:
```

SERVICE -	INTERVAL	BURST
-----	-----	-----
bgp	500	5000



2.2.14.2 Syslog Config Commands

This subsection explains how to configure syslog.

config syslog add

This command is used to add a syslog server to the syslog server list. Note that more than one syslog server can be added in the device.

- Usage:

```
config syslog add <server_address>
```

- Parameters:

- *server_address*: syslog server IP address
- *source*: syslog source IP address
- *port*: syslog server UDP port
- *vrf*: syslog VRF device

- Example:

```
admin@sonic:~$ sudo config syslog add 2.2.2.2 --source 1.1.1.1 --port 514 --vrf default
Running command: systemctl reset-failed rsyslog-config Running
command: systemctl restart rsyslog-config
```

config syslog del

This command is used to delete the configured syslog server.

- Usage:

```
config syslog del <server_address>
```

- Parameters:

- *server_address*: syslog server IP address

- Example:

```
admin@sonic:~$ sudo config syslog del 2.2.2.2
Running command: systemctl reset-failed rsyslog-config Running
command: systemctl restart rsyslog-config
```

config syslog rate-limit-host

This command is used to configure syslog rate limit for host.

- Usage:

```
config syslog rate-limit-host
```

- Parameters:

- *interval*: determines the amount of time that is being measured for rate limiting.
- *burst*: defines the amount of messages, that have to occur in the time limit of interval, to trigger rate limiting

- Example:

```
admin@sonic:~$ sudo config syslog rate-limit-host --interval 300 --burst 20000
```

config syslog rate-limit-container

This command is used to configure syslog rate limit for containers.

- Usage:

```
config syslog rate-limit-container <container-name>
```

- Parameters:

- *interval*: determines the amount of time that is being measured for rate limiting.
- *burst*: defines the amount of messages, that have to occur in the time limit of interval, to trigger rate limiting
- *namespace*: namespace name or all. Value “default” indicates global namespace.



- Example:

```
# Config bgp for all namespaces. For multi ASIC platforms, bgp service in all namespaces will be affected.
# For single ASIC platforms, bgp service in global namespace will be affected. admin@sonic:~$ sudo config
syslog rate-limit-container bgp --interval 300 --burst 20000
# Config bgp for global namespace only.
config syslog rate-limit-container bgp --interval 300 --burst 20000 -n default
# Config bgp for asic0 namespace only.
config syslog rate-limit-container bgp --interval 300 --burst 20000 -n asic0
```

config syslog rate-limit-feature enable

This command is used to enable syslog rate limit feature.

- Usage:

```
config syslog rate-limit-feature enable [<service_name>] -n [<namespace>]
```

- Example:

```
# Enable syslog rate limit for all services in all namespaces
admin@sonic:~$ sudo config syslog rate-limit-feature enable

# Enable syslog rate limit for all services in global namespace
config syslog rate-limit-feature enable -n default
# Enable syslog rate limit for all services in asic0 namespace
config syslog rate-limit-feature enable -n asic0
# Enable syslog rate limit for database in all namespaces
config syslog rate-limit-feature enable database
# Enable syslog rate limit for database in default namespace
config syslog rate-limit-feature enable database -n default
# Enable syslog rate limit for database in asic0 namespace
config syslog rate-limit-feature enable database -n asic0
```

config syslog rate-limit-feature disable

This command is used to disable syslog rate limit feature.

- Usage:

```
config syslog rate-limit-feature disable [<service_name>] -n [<namespace>]
```

- Example:

```
# Disable syslog rate limit for all services in all namespaces
admin@sonic:~$ sudo config syslog rate-limit-feature disable
# Disable syslog rate limit for all services in global namespace
config syslog rate-limit-feature disable -n default
# Disable syslog rate limit for all services in asic0 namespace
config syslog rate-limit-feature disable -n asic0
# Disable syslog rate limit for database in all namespaces
config syslog rate-limit-feature disable database
# Disable syslog rate limit for database in default namespace
config syslog rate-limit-feature disable database -n default
# Disable syslog rate limit for database in asic0 namespace
config syslog rate-limit-feature disable database -n asic0
```



config syslog level

This command is used to configure log level for a given log identifier.

- Usage:

```
config syslog level -i <log_identifier> -l <log_level> --container [<container_name>] --program [<program_name>]
config syslog level -i <log_identifier> -l <log_level> --container [<container_name>] --pid [<process_id>]
config syslog level -i <log_identifier> -l <log_level> ---pid [<process_id>]
```

- Example:

```
# Update the log level without refresh the configuration config syslog level -i xcvrd -l DEBUG
# Update the log level and send SIGHUP to xcvrd running in PMON
config syslog level -i xcvrd -l DEBUG --container pmon --program xcvrd
# Update the log level and send SIGHUP to PID 20 running in PMON
config syslog level -i xcvrd -l DEBUG --container pmon --pid 20
# Update the log level and send SIGHUP to PID 20 running in host
config syslog level -i xcvrd -l DEBUG --pid 20
```

Go Back To Beginning of the document or Beginning of this section

2.2.15 System State

2.2.15.1 Processes

This command is used to determine the CPU utilization. It also lists the active processes along with their corresponding process ID and other relevant parameters.

This sub-section explains the various “processes” specific data that includes the following.

1. cpu Show processes CPU info
2. memory Show processes memory info
3. summary Show processes info

“show processes” commands provide a wrapper over linux’s “top” command. “show process cpu” sorts the processes being displayed by cpu-utilization, whereas “show process memory” does it attending to processes’ memory-utilization.

show processes cpu

This command displays the current CPU usage by process. This command uses linux’s “top -bn 1 -o %CPU” command to display the output.

- Usage:

```
show processes cpu
```

TIP: Users can pipe the output to “head” to display only the “n” number of lines (e.g., show processes cpu | head -n 10)

- Example:

```
admin@sonic:~$ show processes cpu
top - 23:50:08 up 1:18, 1 user, load average: 0.25, 0.29, 0.25
Tasks: 161 total, 1 running, 160 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.8 us, 1.0 sy, 0.0 ni, 95.1 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 8181216 total, 1161060 used, 7020156 free, 105656 buffers
KiB Swap: 0 total, 0 used, 0 free. 557560 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2047 root        20   0 683772 109288 39652 S  23.8  1.3   7:44.79 syncd
 1351 root        20   0 43360   5616  2844 S  11.9  0.1   1:41.56 redis-server
10093 root        20   0 21944   2476  2088 R   5.9  0.0   0:00.03 top
    1 root        20   0 28992   5508  3236 S   0.0  0.1   0:06.42 systemd
    2 root        20   0      0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root        20   0      0      0      0 S   0.0  0.0   0:00.56 ksoftirqd/0
    5 root         0 -20      0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
...
```

TIP: Advanced users can view individual processes using variations of the ps command (e.g., ps -ax | grep <process name>)



show processes memory

This command displays the current memory usage by processes. This command uses linux's "top -bn 1 -o %MEM" command to display the output.

- Usage:

```
show processes memory
```

NOTE that pipe option can be used using " | head -n" to display only the "n" number of lines

- Example:

```
admin@sonic:~$ show processes memory
top - 23:41:24 up 7 days, 39 min, 2 users, load average: 1.21, 1.19, 1.18
Tasks: 191 total, 2 running, 189 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.8 us, 20.7 sy, 0.0 ni, 76.3 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 8162264 total, 5720412 free, 945516 used, 1496336 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 6855632 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 18051 root        20   0  851540 274784  8344 S   0.0   3.4   0:02.77 syncd
 17760 root        20   0 1293428 259212  58732 S   5.9   3.2   96:46.22 syncd
   508 root        20   0  725364  76244  38220 S   0.0   0.9   4:54.49 dockerd
 30853 root        20   0   96348  56824   7880 S   0.0   0.7   0:00.98 show
 17266 root        20   0  509876  49772  30640 S   0.0   0.6   0:06.36 docker
 24891 admin       20   0  515864  49560  30644 S   0.0   0.6   0:05.54 docker
 17643 admin       20   0  575668  49428  30628 S   0.0   0.6   0:06.29 docker
 23885 admin       20   0  369552  49344  30840 S   0.0   0.6   0:05.57 docker
 18055 root        20   0  509076  49260  30296 S   0.0   0.6   0:06.36 docker
 17268 root        20   0  371120  49052  30372 S   0.0   0.6   0:06.45 docker
  1227 root        20   0  443284  48640  30100 S   0.0   0.6   0:41.91 docker
 23785 admin       20   0  443796  48552  30128 S   0.0   0.6   0:05.58 docker
 17820 admin       20   0  435088  48144  29480 S   0.0   0.6   0:06.33 docker
   506 root        20   0 1151040  43140  23964 S   0.0   0.5   8:51.08 containerd
 18437 root        20   0   84852  26388   7380 S   0.0   0.3  65:59.76 python3.6
```

show processes summary

This command displays the current summary information about all the processes

- Usage:

```
show processes summary
```

- Example:

```
admin@sonic:~$ show processes summary
PID  PPID  CMD                                %MEM  %CPU
1      0  /sbin/init                        0.0   0.0
2      0  [kthreadd]                        0.0   0.0
3      2  [ksoftirqd/0]                     0.0   0.0
5      2  [kworker/0:0H]                     0.0   0.0
...
```

[Go Back To Beginning of the document](#) or [Beginning of this section](#)



2.2.16.2 Services & Memory

These commands are used to know the services that are running and the memory that is utilized currently.

show services

This command displays the state of all the SONiC processes running inside a docker container. This helps to identify the status of SONiC's critical processes.

- Usage:

```
show services
```

- Example:

```
admin@sonic:~$ show services
dhcp_relay      docker
-----
UID             PID  PPID  C  STIME TTY      TIME CMD
root             1    0    0  05:26 ?        00:00:12 /usr/bin/python /usr/bin/supervi
root            24    1    0  05:26 ?        00:00:00 /usr/sbin/rsyslogd -n

nat             docker
-----
USER            PID  PPID  C  STIME TTY      TIME CMD
root             1    0    0  05:26 ?        00:00:12 /usr/bin/python /usr/bin/
supervisord
root            18    1    0  05:26 ?        00:00:00 /usr/sbin/rsyslogd -n
root            23    1    0  05:26 ?        00:00:01 /usr/bin/natmgrd
root            34    1    0  05:26 ?        00:00:00 /usr/bin/natsyncd

snmp            docker
-----
UID             PID  PPID  C  STIME TTY      TIME CMD
root             1    0    0  05:26 ?        00:00:16 /usr/bin/python /usr/bin/supervi
root            24    1    0  05:26 ?        00:00:02 /usr/sbin/rsyslogd -n
Debian-+       29    1    0  05:26 ?        00:00:04 /usr/sbin/snmpd -f -LS4d -u Debi
root            31    1    1  05:26 ?        00:15:10 python3.6 -m sonic_ax_impl

syncd           docker
-----
UID             PID  PPID  C  STIME TTY      TIME CMD
root             1    0    0  05:26 ?        00:00:13 /usr/bin/python /usr/bin/supervi
root            12    1    0  05:26 ?        00:00:00 /usr/sbin/rsyslogd -n
root            17    1    0  05:26 ?        00:00:00 /usr/bin/dsserve /usr/bin/syncd
root            27    17  22  05:26 ?        04:09:30 /usr/bin/syncd --diag -p /usr/sh
root            51    27    0  05:26 ?        00:00:01 /usr/bin/syncd --diag -p /usr/sh

swss            docker
-----
UID             PID  PPID  C  STIME TTY      TIME CMD
root             1    0    0  05:26 ?        00:00:29 /usr/bin/python /usr/bin/supervi
root            25    1    0  05:26 ?        00:00:00 /usr/sbin/rsyslogd -n
root            30    1    0  05:26 ?        00:00:13 /usr/bin/orchagent -d /var/log/s
root            42    1    1  05:26 ?        00:12:40 /usr/bin/portsyncd -p /usr/share
root            45    1    0  05:26 ?        00:00:00 /usr/bin/intfsyncd
root            48    1    0  05:26 ?        00:00:03 /usr/bin/neighsyncd
root            59    1    0  05:26 ?        00:00:01 /usr/bin/vlanmgrd
root            92    1    0  05:26 ?        00:00:01 /usr/bin/intfmgrd
root           3606    1    0  23:36 ?        00:00:00 bash -c /usr/bin/arp_update; sle
root           3621  3606    0  23:36 ?        00:00:00 sleep 300
...
```



show system-memory

This command displays the system-wide memory utilization information – just a wrapper over linux native “free” command

- Usage:

```
show system-memory
```

- Example:

```
admin@sonic:~$ show system-memory
```

```
Command: free -m -h
```

	total	used	free	shared	buffers	cached
Mem:	3.9G	2.0G	1.8G	33M	324M	791M
-/+ buffers/cache:		951M	2.9M			
Swap	0B	0B	0B			

show mmu

This command displays virtual address to the physical address translation status of the Memory Management Unit (MMU).

- Usage:

```
show mmu
```

- Example:

```
admin@sonic:~$ show mmu
```

```
Pool: ingress_lossless_pool
```

```
----
Xoff      4194112
type      ingress
mode      dynamic
size      10875072
----
```

```
Pool: egress_lossless_pool
```

```
----
type      egress
mode      static
size      15982720
----
```

```
Pool: egress_lossy_pool
```

```
----
type      egress
mode      dynamic
size      9243812
----
```

```
Profile: egress_lossy_profile
```




```

-----
dynamic_th      3
pool            [BUFFER_POOL|egress_lossy_pool]
size            1518
-----

Profile: pg_lossless_100000_300m_profile
-----
xon_offset      2288
dynamic_th      -3
xon             2288
xoff            268736
pool            [BUFFER_POOL|ingress_lossless_pool]
size            1248
-----

Profile: egress_lossless_profile
-----
static_th       3995680
pool            [BUFFER_POOL|egress_lossless_pool]
size            1518
-----

Profile: pg_lossless_100000_40m_profile
-----
xon_offset      2288
dynamic_th      -3
xon             2288
xoff            177632
pool            [BUFFER_POOL|ingress_lossless_pool]
size            1248
-----

Profile: ingress_lossy_profile
-----
dynamic_th      3
pool            [BUFFER_POOL|ingress_lossless_pool]
size            0
-----

Profile: pg_lossless_40000_40m_profile
-----
xon_offset      2288
dynamic_th      -3
xon             2288
xoff            71552
pool            [BUFFER_POOL|ingress_lossless_pool]
size            1248
-----

```

Go Back To Beginning of the document or Beginning of this section



2.2.16.3 System-Health

These commands are used to monitor the system current running services and hardware state.

show system-health summary

This command displays the current status of 'Services' and 'Hardware' under monitoring. If any of the elements under each of these two sections is 'Not OK' a proper message will appear under the relevant section.

- Usage:

```
show system-health summary
```

- Example:

```
admin@sonic:~$ show system-health summary System status summary
System status LED   red
Services:
  Status: Not OK
  Not Running: 'telemetry', 'sflowmgrd' Hardware:
  Status: OK
```

```
admin@sonic:~$ show system-health summary System status summary
System status LED   green
Services:
  Status: OK
Hardware:
  Status: OK
```

show system-health monitor-list

This command displays a list of all current 'Services' and 'Hardware' being monitored, their status and type.

- Usage:

```
show system-health monitor-list
```

- Example:

```
admin@sonic:~$ show system-health monitor-list System
```

show system-health detail

This command displays the current status of 'Services' and 'Hardware' under monitoring. If any of the elements under each of these two sections is 'Not OK' a proper message will appear under the relevant section. In addition, displays a list of all current 'Services' and 'Hardware' being monitored and a list of ignored elements.

- Usage:

```
show system-health detail
```

- Example:

```
admin@sonic:~$ show system-health detail
```



2.2.17 VLAN & FDB

2.2.17.1 VLAN

2.2.17.1.1 VLAN show commands show

vlan brief

This command displays brief information about all the vlans configured in the device. It displays the vlan ID, IP address (if configured for the vlan), list of vlan member ports, whether the port is tagged or in untagged mode, the DHCPv4 Helper Address, and the proxy ARP status

- Usage:

```
show vlan brief
```

- Example:

```
admin@sonic:~$ show vlan brief
```

show vlan config

This command displays all the vlan configuration.

- Usage:

```
show vlan config
```

- Example:

```
admin@sonic:~$ show vlan config
```

Name	VID	Member	Mode
-----	-----	-----	-----
Vlan100	100	Ethernet0	tagged
Vlan100	100	Ethernet4	tagged

2.2.17.1.2 VLAN Config commands

This sub-section explains how to configure the vlan and its member ports.

config vlan add/del

This command is used to add or delete the vlan.

- Usage:

```
config vlan (add | del) <vlan_id>
```

- Example (Create the VLAN “Vlan100” if it does not already exist):

```
admin@sonic:~$ sudo config vlan add 100
```



config vlan add/del -m

This command is used to add or delete multiple vlans via single command.

- Usage:

```
config vlan (add | del) -m <vlan_id>
```

- Example01 (Create the VLAN “Vlan100, Vlan101, Vlan102, Vlan103” if these does not already exist)
admin@sonic:~\$ sudo config vlan add -m 100-103
- Example02 (Create the VLAN “Vlan105, Vlan106, Vlan107, Vlan108” if these does not already exist):
admin@sonic:~\$ sudo config vlan add -m 105,106,107,108

config vlan member add/del

This command is to add or delete a member port into the already created vlan.

- Usage:

```
config vlan member add/del [-u|--untagged] <vlan_id> <member_portname>
```

NOTE: Adding the -u or --untagged flag will set the member in “untagged” mode

- Example:

```
admin@sonic:~$ sudo config vlan member add 100 Ethernet0 This command
will add Ethernet0 as member of the vlan 100
```

```
admin@sonic:~$ sudo config vlan member add 100 Ethernet4 This command
will add Ethernet4 as member of the vlan 100.
```

config vlan member add/del -m -e This command is to add or delete a member port into multiple already created vlans.

- Usage:

```
config vlan member add/del [-m] [-e] <vlan_id> <member_portname>
```

NOTE: -m flag multiple Vlans in range or comma separted list can be added as a member port. NOTE: -e is used as an except flag as explained with examples below.

- Example:

```
admin@sonic:~$ sudo config vlan member add -m 100-103 Ethernet0
This command will add Ethernet0 as member of the vlan 100, vlan 101, vlan 102, vlan 103
```

```
admin@sonic:~$ sudo config vlan member add -m 100,101,102 Ethernet4
This command will add Ethernet4 as member of the vlan 100, vlan 101, vlan 102
```

```
admin@sonic:~$ sudo config vlan member add -e -m 104,105 Ethernet8
Suppose vlan 100, vlan 101, vlan 102, vlan 103, vlan 104, vlan 105 are existing vlans.
This command will add Ethernet8 as member of  vlan 100, vlan 101, vlan 102, vlan 103
```

```
admin@sonic:~$ sudo config vlan member add -e 100 Ethernet12
Suppose vlan 100, vlan 101, vlan 102, vlan 103, vlan 104, vlan 105 are exisiting vlans.
This command will add Ethernet12 as member of vlan 101, vlan 102, vlan 103, vlan 104, vlan 105
```

```
admin@sonic:~$ sudo config vlan member add all Ethernet20
Suppose vlan 100, vlan 101, vlan 102, vlan 103, vlan 104, vlan 105 are exisiting vlans.
```

```
This command will add Ethernet20 as member of vlan 100, vlan 101, vlan 102, vlan 103, vlan 104, vlan 105
```



****config proxy_arp enabled/disabled****

This command is used to enable or disable proxy ARP for a VLAN interface

- Usage:

config vlan proxy_arp <vlan_id> enabled/disabled

- Example:

```
admin@sonic:~$ sudo config vlan proxy_arp 1000 enabled
```

This command will enable proxy ARP for the interface 'Vlan1000'

Go Back To [Beginning of the document](#) or [Beginning of this section](#vlan--FDB)

FDB

FDB show commands

****show mac****

This command displays the MAC (FDB) entries either in full or partial as given below.

- 1) **show mac** - displays the full table
- 2) **show mac -v <vlanid>** - displays the MACs learnt on the particular VLAN ID.
- 3) **show mac -p <port>** - displays the MACs learnt on the particular port.
- 4) **show mac -a <mac-address>** - display the MACs that match a specific mac-address
- 5) **show mac -t <type>** - display the MACs that match a specific type (static/ dynamic)
- 6) **show mac -c** - display the count of MAC addresses

To show the default MAC address aging time on the switch.

- Usage:

show mac [-v <vlan_id>] [-p <port_name>] [-a <mac_address>] [-t] [-c]

- Example:

```
admin@sonic:~$ show mac
```

No.	Vlan	MacAddress	Port	Type
-----	------	------------	------	------

...

Optionally, you can specify a VLAN ID or interface name or type or mac-address in order to display only that particular entries

- Examples:

```
admin@sonic:~$ show mac -v 1000
```

No.	Vlan	MacAddress	Port	Type
1	1000	BC:22:28:12:34:56	Ethernet19	Dynamic

```
admin@sonic:~$ show mac -p Ethernet19
```

No.	Vlan	MacAddress	Port	Type
1	1000	BC:22:28:12:34:56	Ethernet19	Dynamic

```
admin@sonic:~$ show mac -a BC:22:28:12:34:56
```

No.	Vlan	MacAddress	Port	Type
1	1000	BC:22:28:12:34:56	Ethernet19	Dynamic

```
admin@sonic:~$ show mac -t Static
```

No.	Vlan	MacAddress	Port	Type
2	1000	BC:22:28:12:34:56	Ethernet19	Static
2	1000	BC:22:28:34:56:78	Ethernet19	Static

```
admin@sonic:~$ show mac -c
```

Total number of entries 18



****show mac aging-time****

This command displays the default mac aging time on the switch

```
admin@sonic:~$ show mac aging-time
Aging time for switch is 600 seconds
```

****sonic-clear fdb all****

Clear the FDB table

- Usage:

```
sonic-clear fdb all
```

- Example:

```
admin@sonic:~$ sonic-clear fdb all FDB entries are cleared.
```

VxLAN & Vnet

VxLAN

VxLAN show commands

****show vxlan tunnel****

This command displays brief information about all the vxlans configured in the device. It displays the vxlan tunnel name, source IP address, destination IP address (if configured), tunnel map name and mapping.

- Usage:

```
show vxlan tunnel
```

- Example:

```
admin@sonic:~$ show vxlan tunnel
```

****show vxlan name <vxlan_name>****

This command displays <vxlan_name> configuration.

- Usage:

```
show vxlan name <vxlan_name>
```

- Example:

```
admin@sonic:~$ show vxlan name tunnel3
```

<u>vxlan tunnel name</u>	<u>source ip</u>	<u>destination ip</u>	<u>tunnel map name</u>	<u>tunnel map mapping(vni -> vlan)</u>
tunnel3	10.10.10.10	30.10.10.10	tmap2	1235 -> 200



Vnet

Vnet show commands

****show vnet brief****

This command displays brief information about all the vnets configured in the device. It displays the vnet name, vxlan tunnel name, vni and peer list (if configured).

- Usage:

show vnet brief

- Example:

admin@sonic:~\$ show vnet brief

vnet name	vxlan tunnel	vni	peer list
Vnet_2000	tunnel1	2000	Vnet_2000
Vnet_3000	tunnel1	3000	Vnet4000

****show vnet endpoint <ip/ipv6>****

This command displays the list of vxlan tunnel endpoints and their status. In addition it also shows the number of prefixes associated with each endpoint in the tunnels. If an IP address of an endpoint is provided, this command also shows the associated prefixes as well.

- Usage:

show vnet endpoint <ipv4_address/ipv6_address>

- Example:

admin@sonic:~\$ show vnet endpoint

<u>Endpoint</u>	<u>prefix count</u>	<u>status</u>
fddd:a100:a251::a10:1	1	Down
fddd:a101:a251::a10:1	1	Up

or

admin@sonic:~\$ show vnet endpoint fddd:a101:a251::a10:1

<u>Endpoint</u>	<u>prefix</u>	<u>status</u>
fddd:a101:a251::a10:1	['fddd:a150:a251::a6:1/128']	Up

or

admin@sonic:~\$ show vnet endpoint 100.251.7.1

<u>Endpoint</u>	<u>prefix</u>	<u>status</u>
100.251.7.1	['160.62.191.1/32', '160.63.191.1/32', '160.64.191.1/32']	Up



show vnet name <vnet_name>

This command displays brief information about <vnet_name> configured in the device.

- Usage:

show vnet name <vnet_name>

- Example:

admin@sonic:~\$ show vnet name Vnet_3000

<u>vnet name</u>	<u>vxlan tunnel</u>	<u>vni</u>	<u>peer list</u>
Vnet_3000	tunnel1	3000	Vnet_2000,Vnet4000

show vnet interfaces

This command displays vnet interfaces information about all the vnets configured in the device.

- Usage:

show vnet interfaces

- Example:

admin@sonic:~\$ show vnet interfaces

<u>vnet name</u>	<u>interfaces</u>
Vnet_2000	Ethernet1
Vnet_3000	Vlan2000

show vnet neighbors

This command displays vnet neighbor information about all the vnets configured in the device. It displays the vnet name, neighbor IP address, neighbor mac address (if configured) and interface.

- Usage:

show vnet neighbors

- Example:

admin@sonic:~\$ show vnet neighbors Vnet_2000 neighbor mac_address interfaces

11.11.11.11	Ethernet1
11.11.11.12	Ethernet2
Vnet_3000 neighbor mac_address interfaces	
20.20.20.20	aa:bb:cc:dd:ee:ff Vlan2000

show vnet routes all

This command displays all routes information about all the vnets configured in the device. It also show the vnet routes which are configured but may or may not be active based on endpoint BFD status.

- Usage:

show vnet routes all

- Example:

admin@sonic:~\$ show vnet routes all

<u>vnet name</u>	<u>prefix</u>	<u>nexthop</u>	<u>interface</u>
Vnet_2000	100.100.3.0/24	Ethernet52	
Vnet_3000	100.100.4.0/24	Vlan2000	

<u>vnet name</u>	<u>prefix</u>	<u>endpoint</u>	<u>mac address</u>	<u>vni status</u>
Vnet_2000	100.100.1.1/32	10.10.10.1	00:00:00:00:03:04	active
Vnet_3000	100.100.2.1/32	10.10.10.2	00:00:00:00:03:04	inactive
Vnet_3000	100.100.2.3/32	10.10.10.6	00:00:00:00:03:04	

show vnet routes tunnel

This command displays tunnel routes information about all the vnets configured in the device.

- Usage:

show vnet routes tunnel

- Example:

admin@sonic:~\$ show vnet routes tunnel

<u>vnet name</u>	<u>prefix</u>	<u>endpoint</u>	<u>mac address</u>	<u>vni</u>
...				



Warm Reboot

warm-reboot command initiates a warm reboot of the device.

warm-reboot command doesn't require setting warm restart configuration. The command will setup everything needed to perform warm reboot.

This command requires root privilege.

- Usage:

warm-reboot [-h|-?|-v|-f|-r|-k|-x|-c |-s|-D]

- Parameters:

- h,-? : get this help
- v : turn on verbose mode
- f : force execution
- r : reboot with /sbin/reboot
- k : reboot with /sbin/kexec -e [default]
- x : execute script with -x flag
- c : specify control plane assistant IP list
- s : strict mode: do not proceed without:
 - control plane assistant IP list.
- D : detached mode - closing terminal will not cause stopping reboot

- Example:

admin@sonic:~\$ sudo warm-reboot -v

Warm Restart

Besides device level warm reboot, SONiC also provides docker based warm restart. This feature is currently supported by following dockers: BGP, teamD, and SWSS. A user can manage to restart a particular docker, with no interruption on packet forwarding and no effect on other services. This helps to reduce operational costs as well as development efforts. For example, to fix a bug in BGP routing stack, only the BGP docker image needs to be built, tested and upgraded.

To achieve uninterrupted packet forwarding during the restarting stage and database reconciliation at the post restarting stage, warm restart enabled dockers with adjacency state machine facilitate standardized protocols.

For example, a BGP restarting switch **must** have BGP "Graceful Restart" enabled, and its BGP neighbors must be "Graceful Restart Helper Capable", as specified in [IETF RFC4724](https://tools.ietf.org/html/rfc4724).

Before warm restart BGP docker, the following BGP commands should be enabled:

bgp graceful-restart bgp graceful-restart preserve-fw-state

In current SONiC release, the above two commands are enabled by default.

It should be aware that during a warm restart, certain BGP fast convergence feature and black hole avoidance feature should either be disabled or be set to a lower preference to avoid conflicts with BGP graceful restart.

For example, BGP BFD could be disabled via:

no neighbor <A.B.C.D|X:X::X:X|WORD> bfd

otherwise, the fast failure detection would cause packet drop during warm reboot.

Another commonly deployed blackhole avoidance feature: dynamic route priority adjustment, could be disabled via:

no bgp max-med on-peerup

to avoid large routes churn during BGP restart.

Warm Restart show commands



****show warm_restart config****

This command displays all the configuration related to warm_restart.

- Usage:

show warm_restart config

- Example:

admin@sonic:~\$ show warm_restart config

<u>name</u>	<u>enable</u>	<u>timer_name</u>	<u>timer_duration</u>
bgp	true	bgp_timer	100
teamd	false	teamsyncd_timer	300
swss	false	neighsyncd_timer	200
system	true	NULL	NULL

****show warm_restart state****

This command displays the warm_restart state.

- Usage:

show warm_restart state

- Example:

admin@sonic:~\$ show warm_restart state

<u>name</u>	<u>restore_count</u>	<u>state</u>
orchagent	0	
vlanmgrd	0	
bgp	1	
reconciled portsyncd	0	
teammgrd	1	
neighsyncd	0	
teamsyncd	1	
syncd	0	
natsyncd	0	



Warm Restart Config commands

This sub-section explains the various configuration related to warm restart feature.

Following parameters can be configured using this command.

- 1) bgp_timer
- 2) disable
- 3) enable
- 4) neighsyncd_timer
- 5) teamsyncd_timer

Each of these sub-commands are explained in the following section.

Users can use an optional parameter "-s" to use the unix domain socket for communicating with the RedisDB which will be faster when compared to using the default network sockets.

All these commands have the following option.

Options:

-s, --redis-unix-socket-path TEXT
unix socket path for redis connection

config warm_restart bgp_timer

This command is used to set the bgp_timer value for warm_restart of BGP service. bgp_timer holds the time interval utilized by fpmSyncd during warm-restart episodes.

During this interval fpmSyncd will recover all the routing state previously pushed to AppDB, as well as all the new state coming from zebra/bgpd.

Upon expiration of this timer, fpmSyncd will execute the reconciliation logic to eliminate all the stale entries from AppDB.

This timer should match the BGP-GR restart-timer configured within the elected routing-stack.

Supported range: 1-3600.

- Usage:

config warm_restart [-s|--redis-unix-socket-path <socket_path>] bgp_timer

- Parameters:

- seconds: Range from 1 to 3600

- Example:

admin@sonic:~\$ sudo config warm_restart bgp_timer 1000



****config warm_restart enable/disable****

This command is used to enable or disable the warm_restart for a particular service that supports warm reboot.

Following four services support warm reboot. When user restarts the particular service using "systemctl restart <service_name>", this configured value will be checked for whether it is enabled or disabled.

If this configuration is enabled for that service, it will perform warm reboot for that service. Otherwise, it will do cold restart of the service.

- Usage:

```
config warm_restart [-s|--redis-unix-socket-path <socket_path>] enable [<module_name>]
```

- Parameters:

- module_name: Can be either system or swss or bgp or teamd. If "module_name" argument is not specified, it will enable "system" module.

- Example (Set warm_restart as "enable" for the "system" service):

```
admin@sonic:~$ sudo config warm_restart enable
```

- Example (Set warm_restart as "enable" for the "swss" service. When user does "systemctl restart swss", it will perform warm reboot instead of cold reboot)

```
admin@sonic:~$ sudo config warm_restart enable swss
```

- Example (Set warm_restart as "enable" for the "teamd" service. When user does "systemctl restart teamd", it will perform warm reboot instead of cold reboot)

```
admin@sonic:~$ sudo config warm_restart enable teamd
```

****config warm_restart neighsyncd_timer****

This command is used to set the neighsyncd_timer value for warm_restart of "swss" service.

neighsyncd_timer is the timer used for "swss" (neighsyncd) service during the warm restart.

Timer is started after the neighborTable is restored to internal data structures. neighborsyncd then starts to read all Linux kernel entries and mark the entries in the data structures accordingly.

Once the timer is expired, reconciliation is done and the delta is pushed to appDB Valid value is 1-9999. 0 is invalid.

- Usage:

```
config warm_restart [-s|--redis-unix-socket-path <socket_path>] neighsyncd_timer
```

- Parameters:

- seconds: Range from 1 to 9999

- Example:

```
admin@sonic:~$ sudo config warm_restart neighsyncd_timer 2000
```

****config warm_restart bgp_timer****

This command is used to set the bgp_timer value for warm_restart of "bgp" service. bgp_timer is the timer used for "bgp" service during the warm restart.

Timer is started after the BGP table is restored to internal data structures. BGP services then start to read all Linux kernel entries and mark the entries in the data structures accordingly.

Once the timer is expired, reconciliation is done and the delta is pushed to appDB Valid value is 1-9999. 0 is invalid.

- Usage:

```
config warm_restart [-s|--redis-unix-socket-path <socket_path>] bgp_timer
```

- Parameters:

- seconds: Range from 1 to 9999

- Example:

```
admin@sonic:~$ sudo config warm_restart bgp_timer 2000
```



****config warm_restart teamsyncd_timer****

This command is used to set the teamsyncd_timer value for warm_restart of teamd service.

teamsyncd_timer holds the time interval utilized by teamsyncd during warm-restart episodes.

The timer is started when teamsyncd starts. During the timer interval, teamsyncd will preserve all LAG interface changes, but it will not apply them.

The changes will only be applied when the timer expires.

When the changes are applied, the stale LAG entries will be removed, the new LAG entries will be created.

Supported range: 1-9999. 0 is invalid

- Usage:

config warm_restart teamsyncd_timer

- Parameters:

- seconds: Range from 1 to 9999

- Example:

admin@sonic:~\$ sudo config warm_restart teamsyncd_timer 3000

Watermark

Watermark Show commands

****show watermark telemetry interval****

This command displays the configured interval for the telemetry.

- Usage:

show watermark telemetry interval

- Example:

admin@sonic:~\$ show watermark telemetry interval

Telemetry interval 120 second(s)

Watermark Config commands

****config watermark telemetry interval****

This command is used to configure the interval for telemetry. The default interval is 120 seconds.

There is no regulation on the valid range of values; it leverages linux timer.

- Usage:

config watermark telemetry interval

- Example:

admin@sonic: ~\$ sudo config watermark telemetry interval 999



Software Installation and Management

SONiC images can be installed in one of two methods:

1. From within a running SONiC image using the **`sonic-installer`** utility
2. From the vendor's bootloader (E.g., ONIE, About, etc.)

SONiC packages are available as prebuilt Docker images and meant to be installed with the ***sonic-package-manager*** utility.

SONiC Package Manager

The ***sonic-package-manager*** is a command line tool to manage (e.g. install, upgrade or uninstall) SONiC Packages.

sonic-package-manager list

This command lists all available SONiC packages, their description, installed version and installation status.

SONiC package status can be ***Installed***, ***Not installed*** or ***Built-In***. "Built-In" status means that a feature is built-in to SONiC image and can't be upgraded or uninstalled.

- Usage:

sonic-package-manager list

- Example:

```
admin@sonic:~$ sonic-package-manager list
```

Name	Repository	Description	Version	Status
cpu-report	azure/cpu-report	CPU report package	N/A	Not Installed
database	docker-database	SONiC database package	1.0.0	Built-In
dhcp-relay	azure/docker-dhcp-relay	SONiC dhcp-relay package	1.0.0	Installed
fpm-frr	docker-fpm-frr	SONiC fpm-frr package	1.0.0	Built-In
lldp	docker-lldp	SONiC lldp package	1.0.0	Built-In
macsec	docker-macsec	SONiC macsec package	1.0.0	Built-In
mgmt-framework	docker-sonic-mgmt-framework	SONiC mgmt-framework package	1.0.0	Built-In
nat	docker-nat	SONiC nat package	1.0.0	Built-In
pmon	docker-platform-monitor	SONiC pmon package	1.0.0	Built-In
radv	docker-router-advertiser	SONiC radv package	1.0.0	Built-In
sflow	docker-sflow	SONiC sflow package	1.0.0	Built-In
snmp	docker-snmp	SONiC snmp package	1.0.0	Built-In
swss	docker-orchagent	SONiC swss package	1.0.0	Built-In
teamd	docker-teamd	SONiC teamd package	1.0.0	Built-In
telemetry	docker-sonic-telemetry	SONiC telemetry package	1.0.0	Built-In

sonic-package-manager repository add

This command will add a new repository as source for SONiC packages to the database. ***NOTE***: requires elevated (root) privileges to run

- Usage:

Usage: **sonic-package-manager repository add [OPTIONS] NAME REPOSITORY**

Add a new repository to database.

NOTE: This command requires elevated (root) privileges to run.



Options:

- default-reference TEXT Default installation reference. Can be a tag or sha256 digest in repository.
- description TEXT Optional package entry description.
- help Show this message and exit.

- Example:

```
admin@sonic:~$ sudo sonic-package-manager repository add
cpu-report azure/sonic-cpu-report --default-reference 1.0.0
```

****sonic-package-manager repository remove****

This command will remove a repository as source for SONiC packages from the database . The package has to be **Not Installed** in order to be removed from package database. **NOTE**: requires elevated (root) privileges to run

- Usage:

Usage: sonic-package-manager repository remove [OPTIONS] NAME

Remove repository from database.

NOTE: This command requires elevated (root) privileges to run.

Options: -help Show this message and exit.

- Example:

```
admin@sonic:~$ sudo sonic-package-manager repository remove cpu-report
```

****sonic-package-manager install****

This command pulls and installs a package on SONiC host. **NOTE**: this command requires elevated (root) privileges to run

- Usage:

Usage: sonic-package-manager install [OPTIONS] [PACKAGE_EXPR]

Install/Upgrade package using [PACKAGE_EXPR] in format "<name>[=<version>|@<reference>]".

The repository to pull the package from is resolved by lookup in package database, thus the package has to be added via "sonic-package-manager repository add" command.

In case when [PACKAGE_EXPR] is a package name "<name>" this command will install or upgrade to a version referenced by "default-reference" in package database.

NOTE: This command requires elevated (root) privileges to run.



Options:

–enable Set the default state of the feature to enabled and enable feature right after installation.

NOTE: user needs to execute “config save -y” to make this setting persistent.

–set-owner [local|kube] Default owner configuration setting for a feature.

–from-repository TEXT Fetch package directly from image registry repository.

NOTE: This argument is mutually exclusive with arguments: [package_expr, from_tarball].

–from-tarball FILE Fetch package from saved image tarball.

NOTE: This argument is mutually exclusive with arguments: [package_expr, from_repository].

–f, –force Force operation by ignoring package dependency tree and package manifest validation failures.

–y, –yes Automatically answer yes on prompts.

–v, –verbosity LVL Either CRITICAL, ERROR, WARNING, INFO or DEBUG. Default is INFO.

–skip-host-plugins Do not install host OS plugins provided by the package (CLI, etc).

NOTE: In case when package host OS plugins are set as mandatory in package manifest this option will fail the installation.

–allow-downgrade Allow package downgrade. By default an attempt to downgrade the package will result in a failure since downgrade might not be supported by the package, thus requires explicit request from the user.

–help Show this message and exit..

- Example:

```
admin@sonic:~$ sudo sonic-package-manager install dhcp-relay=1.0.2
admin@sonic:~$ sudo sonic-package-manager install dhcp-relay@latest
admin@sonic:~$ sudo sonic-package-manager install dhcp-re-
lay@sha256:9780f6d83e45878749497a6297ed9906c19ee0cc48cc88dc63827564bb8768fd
admin@sonic:~$ sudo sonic-package-manager install –from-repository azure/sonic-cpu-report:latest
admin@sonic:~$ sudo sonic-package-manager install –from-tarball sonic-docker-image.gz
```

****sonic-package-manager uninstall****

This command uninstalls package from SONiC host. User needs to stop the feature prior to uninstalling it.

NOTE: this command requires elevated (root) privileges to run.

- Usage:

Usage: sonic-package-manager uninstall [OPTIONS] NAME

Uninstall package.

NOTE: This command requires elevated (root) privileges to run.

Options:

–f, –force Force operation by ignoring package dependency tree and package manifest validation failures.

–y, –yes Automatically answer yes on prompts.

–v, –verbosity LVL Either CRITICAL, ERROR, WARNING, INFO or DEBUG. Default is INFO.

–help Show this message and exit.

- Example:

```
admin@sonic:~$ sudo sonic-package-manager uninstall dhcp-relay
```




****sonic-package-manager reset****

This command resets the package by reinstalling it to its default version. *NOTE*: this command requires elevated (root) privileges to run.

- Usage:

Usage: sonic-package-manager reset [OPTIONS] NAME

Reset package to the default version.

NOTE: This command requires elevated (root) privileges to run.

Options:

-f, --force Force operation by ignoring package dependency tree and package manifest validation failures.

-y, --yes Automatically answer yes on prompts.

-v, --verbosity LVL Either CRITICAL, ERROR, WARNING, INFO or DEBUG. Default is INFO.

--skip-host-plugins Do not install host OS plugins provided by the package (CLI, etc).

NOTE: In case when package host OS plugins are set as mandatory in package manifest this option will fail the installation.

--help Show this message and exit.

- Example:

admin@sonic:~\$ sudo sonic-package-manager reset dhcp-relay

****sonic-package-manager show package versions****

This command will retrieve a list of all available versions for the given package from the configured upstream repository

- Usage:

Usage: sonic-package-manager show package versions [OPTIONS] NAME

Show available versions.

Options:

--all Show all available tags in repository.

--plain Plain output.

--help Show this message and exit.

- Example:

admin@sonic:~\$ sonic-package-manager show package versions dhcp-relay

• 1.0.0 • 1.0.2 • 2.0.0

admin@sonic:~\$ sonic-package-manager show package versions dhcp-relay --plain

• 1.0.0 • 1.0.2 • 2.0.0

admin@sonic:~\$ sonic-package-manager show package versions dhcp-relay --all

• 1.0.0 • 1.0.2 • 2.0.0 • latest



****sonic-package-manager show package changelog****

This command fetches the changelog from the package manifest and displays it. **NOTE**: package changelog can be retrieved from registry or read from image tarball without installing it.

- Usage:

Usage: sonic-package-manager show package changelog [OPTIONS] [PACKAGE_EXPR]

Show package changelog.

Options:

-from-repository TEXT Fetch package directly from image registry repository

NOTE: This argument is mutually exclusive with arguments: [from_tarball, package_expr].

-from-tarball FILE Fetch package from saved image tarball

NOTE: This argument is mutually exclusive with arguments: [package_expr, from_repository].

-help Show this message and exit.

- Example:

```
admin@sonic:~$ sonic-package-manager show package changelog dhcp-relay 1.0.0:
```

- Initial release

Author (author@email.com) Mon, 25 May 2020 12:25:00 +0300

****sonic-package-manager show package manifest****

This command fetches the package manifest and displays it. **NOTE**: package manifest can be retrieved from registry or read from image tarball without installing it.

- Usage:

Usage: sonic-package-manager show package manifest [OPTIONS] [PACKAGE_EXPR]

Show package manifest.

Options:

-from-repository TEXT Fetch package directly from image registry repository

NOTE: This argument is mutually exclusive with arguments: [package_expr, from_tarball].

-from-tarball FILE Fetch package from saved image tarball

NOTE: This argument is mutually exclusive with arguments: [from_repository, package_expr].

-v, -verbosity LVL Either CRITICAL, ERROR, WARNING, INFO or DEBUG

-help Show this message and exit.

- Example:

```
admin@sonic:~$ sonic-package-manager show package manifest dhcp-relay=2.0.0 { "version": "1.0.0",
"package": { "version": "2.0.0", "depends": [ "database>=1.0.0,<2.0.0" ] }, "service": { "name":
"dhcp_relay" } }
```



SONiC Installer

This is a command line tool available as part of the SONiC software; If the device is already running the SONiC software, this tool can be used to install an alternate image in the partition.

This tool has facility to install an alternate image, list the available images and to set the next reboot image.

This command requires elevated (root) privileges to run.

sonic-installer list

This command displays information about currently installed images. It displays a list of installed images, currently running image and image set to be loaded in next reboot.

- Usage:

sonic-installer list

- Example:

```
admin@sonic:~$ sudo sonic-installer list
Current: SONiC-OS-HEAD.XXXX
Next: SONiC-OS-HEAD.XXXX
Available: SONiC-OS-HEAD.XXXX SONiC-OS-HEAD.YYYY
```

TIP: This output can be obtained without elevated privileges by running the `show boot` command. See [\[here\]\(#show-system-status\)](#) for details.

sonic-installer install

This command is used to install a new image on the alternate image partition. This command takes a path to an installable SONiC image or URL and installs the image.

- Usage:

```
sonic-installer install <image_file_path>
```



- Example:

```

```bash
admin@sonic:~$ sudo sonic-installer install "https://sonic-build.azurewebsites.
net/api/sonic/artifacts?branchName=xxxx&platform=xxxx&target=target%2Fsonic-xxxx. bin"
New image will be installed, continue? [y/N]: y
Downloading image...
...99%, 980 MB, 1017 KB/s, 0 seconds left...
Installing image SONiC-OS-xxxxx and setting it as default...

Command: bash /tmp/sonic_image
Verifying image checksum ... OK.
Preparing image archive ... OK.
Installing SONiC in SONiC
ONIE Installer: platform: x86_64-vs-r0
onie_platform: x86_64-kvm_x86_64-r0 Installing
SONiC to /host/image-xxxx Archive: fs.zip
 creating: /host/image-xxxx/boot/
 inflating: /host/image-xxxx/boot/System.map-5.10.0-12-2-amd64 inflating:
 /host/image-xxxx/boot/initrd.img-5.10.0-12-2-amd64 inflating: /host/image-
 xxxx/boot/vmlinuz-5.10.0-12-2-amd64 inflating: /host/image-xxxx/boot/config-
 5.10.0-12-2-amd64
extracting: /host/image-xxxx/fs.squashfs
ONIE_IMAGE_PART_SIZE=32768
EXTRA_CMDLINE_LINUX=
Switch CPU vendor is: GenuineIntel Switch
CPU cstates are: disabled
EXTRA_CMDLINE_LINUX=
Installed SONiC base image SONiC-OS successfully

Command: grub-set-default --boot-directory=/host 0

Command: config-setup backup
Taking backup of current configuration

Command: mkdir -p /tmp/image-xxxx-fs
Command: mount -t squashfs /host/image-xxxx/fs.squashfs /tmp/image-xxxx-fs Command: sonic-cfggen -d
-y /tmp/image-xxxx-fs/etc/sonic/sonic_version.yml -t /tmp/image-xxxx-fs/usr/share/sonic/templates/sonic-
environment.j2
Command: umount -r -f /tmp/image-xxxx-fs
Command: rm -rf /tmp/image-xxxx-fs Command:
mkdir -p /tmp/image-xxxx-fs
Command: mount -t squashfs /host/image-xxxx/fs.squashfs /tmp/image-xxxx-fs Command: mkdir -p
/host/image-xxxx/rw
Command: mkdir -p /host/image-xxxx/work
Command: mkdir -p /tmp/image-xxxx-fs
Command: mount overlay -t overlay -o rw,relatime,lowerdir=/tmp/image-xxxx-fs,upperdir=/host/image-
xxxx/rw,workdir=/host/image-xxxx/work /tmp/image-xxxx-fs Command: mkdir -p /tmp/image-xxxx-
fs/var/lib/docker
Command: mount --bind /host/image-xxxx/docker /tmp/image-xxxx-fs/var/lib/docker Command: chroot
/tmp/image-xxxx-fs mount proc /proc -t proc
Command: chroot /tmp/image-xxxx-fs mount sysfs /sys -t sysfs
Command: cp /tmp/image-xxxx-fs/etc/default/docker /tmp/image-xxxx-fs/tmp/

```



```

docker_config_backup
Command: sh -c echo 'DOCKER_OPTS="$DOCKER_OPTS -H unix:// --storage-driver=overlay2 --
bip=240.127.1.1/24 --iptables=false --ipv6=true --fixed-cidr-v6=fd00::/80 "' >> /tmp/image-xxxx-
fs/etc/default/docker
Command: chroot /tmp/image-xxxx-fs /usr/lib/docker/docker.sh start mount: /sys/fs/cgroup/cpu:
cgroup already mounted on /sys/fs/cgroup. mount: /sys/fs/cgroup/cpuacct: cgroup already mounted
on /sys/fs/cgroup.
Command: cp /var/lib/sonic-package-manager/packages.json /tmp/image-xxxx-fs/tmp/ packages.json
Command: touch /tmp/image-xxxx-fs/tmp/docker.sock
Command: mount --bind /var/run/docker.sock /tmp/image-xxxx-fs/tmp/docker.sock Command: chroot
/tmp/image-xxxx-fs sh -c command -v sonic-package-manager Command: chroot /tmp/image-xxxx-fs
sonic-package-manager migrate /tmp/ packages.json --dockerd-socket /tmp/docker.sock -y
migrating package dhcp-relay
skipping dhcp-relay as installed version is newer migrating package
macsec
skipping macsec as installed version is newer
Command: chroot /tmp/image-xxxx-fs /usr/lib/docker/docker.sh stop
Command: mv /tmp/image-xxxx-fs/tmp/docker_config_backup /tmp/image-xxxx-fs/etc/ default/docker
Command: umount -f -R /tmp/image-xxxx-fs
Command: umount -r -f /tmp/image-xxxx-fs
Command: rm -rf /tmp/image-xxxx-fs Command:
sync;sync;sync

Command: sleep 3

Done

```

Installing a new image using the sonic-installer will keep using the packages installed on the currently running SONiC image and automatically migrate those. In order to perform clean SONiC installation use the *--skip-package-migration* option:

- Example:

```

admin@sonic:~$ sudo sonic-installer install "https://sonic-build.azurewebsites.
net/api/sonic/artifacts?branchName=xxxx&platform=xxxx&target=target%2Fsonic-xxxx.bin" --skip-
package-migration

```

## sonic-installer set\_default

This command is be used to change the image which can be loaded by default in all the subsequent reboots.

- Usage:

```

sonic-installer set_default <image_name>

```

- Example:

```

admin@sonic:~$ sudo sonic-installer set_default SONiC-OS-HEAD.XXXX

```

## sonic-installer set\_next\_boot

This command is used to change the image that can be loaded in the *next* reboot only. Note that it will fallback to current image in all other subsequent reboots after the next reboot.

- Usage:

```

sonic-installer set_next_boot <image_name>

```

- Example:

```

admin@sonic:~$ sudo sonic-installer set_next_boot SONiC-OS-HEAD.XXXX

```



### sonic-installer remove

This command is used to remove the unused SONiC image from the disk. Note that it's *not* allowed to remove currently running image.

- Usage:

```
sonic-installer remove [-y|--yes] <image_name>
```

- Example:

```
admin@sonic:~$ sudo sonic-installer remove SONiC-OS-HEAD.YYYY Image will be
removed, continue? [y/N]: y
Updating GRUB...
Done
Removing image root filesystem... Done
Command: grub-set-default --boot-directory=/host 0

Image removed
```

### sonic-installer cleanup

This command removes all unused images from the device, leaving only the currently active image and the image which will be booted into next (if different) installed. If there are no images which can be removed, the command will output `No image(s) to remove`

- Usage:

```
sonic-installer cleanup [-y|--yes]
```

- Example:

```
admin@sonic:~$ sudo sonic-installer cleanup
Remove images which are not current and next, continue? [y/N]: y
No image(s) to remove
```

### sonic-installer get-fips

This command is used to get the fips status.

- Usage:

```
sonic-installer get-fips [<image_name>]
```

If the image is not specified, the next boot image will be used.

- Example:

```
admin@sonic:~$ sudo sonic-installer get-fips FIPS is enabled
```

### sonic-installer set-fips

The command is used to enable or disable the fips.

- Usage:

```
sonic-installer set-fips [--enable-fips|--disable-fips] [<image_name>]
```

If the option `--enable-fips` or `--disable-fips` is not specified, the default behavior is to enable fips.

- Example:

```
admin@sonic:~$ sudo sonic-installer set-fips --enable-fips Done
Set FIPS for the image successfully
```



### 2.2.18 Troubleshooting Commands

For troubleshooting and debugging purposes, this command gathers pertinent information about the state of the device; information is as diverse as syslog entries, database state, routing-stack state, etc., It then compresses it into an archive file. This archive file can be sent to the SONiC development team for examination. Resulting archive file is saved as `/var/dump/<DEVICE_HOST_NAME>_YYYYMMDD_HHMMSS.tar.gz`

- Usage:

```
show techsupport
```

- Example:

```
admin@sonic:~$ show techsupport [--since=<time_specifier>]
```

If the SONiC system was running for quite some time `show techsupport` will produce a large dump file.

To reduce the amount of syslog and core files gathered during system dump use `--since` option:

- Examples:

```
admin@sonic:~$ show techsupport --since=yesterday # Will collect syslog and core files for the last 24 hours
```

```
admin@sonic:~$ show techsupport --since='hour ago' # Will collect syslog and core files for the last one hour
```



### 2.2.18.1 Debug Dumps

In SONiC, there usually exists a set of tables related/relevant to a particular module. All of these might have to be looked at to confirm whether any configuration update is properly applied and propagated. This utility comes in handy because it prints a unified view of the redis-state for a given module

- Usage:

```
Usage: dump state [OPTIONS] MODULE IDENTIFIER
```

Dump the redis-state of the identifier for the module specified

Options:

-s, --show	Display Modules Available
-d, --db TEXT	Only dump from these Databases
-t, --table	Print in tabular format [default: False]
-k, --key-map	Only fetch the keys matched, don't extract field-value dumps [default: False]
-v, --verbose	Prints any intermediate output to stdout useful for dev & troubleshooting [default: False]
-n, --namespace TEXT	Dump the redis-state for this namespace. [default: DEFAULT_NAMESPACE]
--help	Show this message and exit.

- Examples:

```
root@sonic# dump state --show
Module Identifier
-----port -----
copp port_name
 trap_id
```

```
admin@sonic:~$ dump state copp arp_req --key-map --db ASIC_DB
{"arp_req":
 { "ASIC_DB":
 { "keys":
 [
 "ASIC_STATE:SAI_OBJECT_TYPE_HOSTIF_TRAP:oid:0x22000000000c5b",
 "ASIC_STATE:SAI_OBJECT_TYPE_HOSTIF_TRAP_GROUP:oid:0x11000000000c59",
 "ASIC_STATE:SAI_OBJECT_TYPE_POLICER:oid:0x12000000000c5a",
 "ASIC_STATE:SAI_OBJECT_TYPE_QUEUE:oid:0x15000000000626"
],
 "tables_not_found": [],
 "vidtorid":
 {
 "oid:0x22000000000c5b": "oid:0x200000000022",
 "oid:0x11000000000c59": "oid:0x300000011",
 "oid:0x12000000000c5a": "oid:0x200000012",
 "oid:0x15000000000626": "oid:0x12e0000040015"
 }
 }
 }
 }
}
```





### 2.2.18.2 Event Driven Techsupport Invocation

This feature/capability makes the techsupport invocation event-driven based on system events like core dump generation or low RAM availability. This feature is only applicable for the processes running in the containers. More detailed explanation can be found in the HLD [https://github.com/sonic-net/SONiC/blob/master/doc/auto\\_techsupport\\_and\\_coredump\\_mgmt.md](https://github.com/sonic-net/SONiC/blob/master/doc/auto_techsupport_and_coredump_mgmt.md)

#### 2.2.18.2.1 config auto-techsupport global commands

##### config auto-techsupport global state

- Usage:  

```
config auto-techsupport global state <enabled/disabled>
```
- Example:  

```
config auto-techsupport global state enabled
```

##### config auto-techsupport global rate-limit-interval

- Usage:  

```
config auto-techsupport global rate-limit-interval <uint16>
```
- Parameters:
  - rate-limit-interval: Minimum time in seconds to wait after the last techsupport creation time before invoking a new one.
- Example:  

```
config auto-techsupport global rate-limit-interval 200
```

##### config auto-techsupport global max-techsupport-limit

- Usage:  

```
config auto-techsupport global max-techsupport-limit <float upto two decimal places>
```
- Parameters:
  - max-techsupport-limit: A percentage value should be specified. This signifies maximum size to which /var/dump/ directory can be grown until.
- Example:  

```
config auto-techsupport global max-techsupport-limit 10.15
```

##### config auto-techsupport global max-core-limit

Usage:

- ```
config auto-techsupport global max-core-limit <float upto two decimal places>
```
- Parameters:
 - max-core-limit: A percentage value should be specified. This signifies maximum size to which /var/core/ directory can be grown until.
 - Example:

```
config auto-techsupport global max-core-limit 10.15
```



config auto-techsupport global since

- Usage:

```
config auto-techsupport global since <string>
```

- Parameters:

- since: This limits the auto-invoked techsupport to only collect the logs & core-dumps generated since the time provided. Any valid date string of the formats specified here can be used. (https://www.gnu.org/software/coreutils/manual/html_node/Date-input-formats.html). If this value is not explicitly configured or a non-valid string is provided, a default value of “2 days ago” is used.

- Example:

```
config auto-techsupport global since <string>
```

config auto-techsupport global available-mem-threshold

Configure available memory threshold in %. System will automatically generate a techsupport dump when available memory goes below the configured threshold. Setting this field to 0 will disable techsupport invocation.

- Usage:

```
config auto-techsupport global available-mem-threshold <float up to two decimal places>
```

- Parameters:

- available-mem-threshold: Memory threshold. Configure 0 to disable techsupport invocation on memory usage threshold crossing

- Example:

```
config auto-techsupport global available-mem-threshold 20
```

config auto-techsupport global min-available-mem

Configure minimum available memory in MB. System will automatically generate a techsupport dump when available memory goes below the configured threshold.

- Usage:

```
config auto-techsupport global min-available-mem <uint32>
```

- Parameters:

- min-available-mem: Minimum free memory amount in MB to trigger techsupport dump

- Example:

```
config auto-techsupport global min-available-mem 200
```

2.2.18.2.2 config auto-techsupport-feature commands

config auto-techsupport-feature add

- Usage:

```
config auto-techsupport-feature add <feature_name> --state <enabled/disabled> --rate-limit-interval <uint16> --available-mem-threshold <float up to two decimal places>
```

- Parameters:

- state: enable/disable the capability for the specific feature/container.
- rate-limit-interval: Rate limit interval for the corresponding feature. Configure 0 to explicitly disable. For the techsupport to be generated by auto-techsupport, both the global and feature specific rate-limit-interval has to be passed
- available-mem-threshold: Memory threshold. Configure 0 to disable techsupport invocation on memory usage threshold crossing.

- Example:

```
config auto-techsupport-feature add bgp --state enabled --rate-limit-interval 200 --available-mem-threshold 50
```



config auto-techsupport-feature delete

- Usage:

```
config auto-techsupport-feature delete <feature_name>
```

- Example:

```
config auto-techsupport-feature delete swss
```

config auto-techsupport-feature update

- Usage:

```
config auto-techsupport-feature update <feature_name> --state <enabled/disabled> config auto-techsupport-  
feature update <feature_name> --rate-limit-interval <uint16>  
config auto-techsupport-feature update <feature_name> --available-mem-threshold <float up to two  
decimal places>
```

- Example:

```
config auto-techsupport-feature update snmp --state enabled  
config auto-techsupport-feature update swss --rate-limit-interval 200
```

2.2.18.2.3 show auto-techsupport commands

show auto-techsupport global

- Usage:

```
show auto-techsupport global
```

- Example:

```
admin@sonic:~$ show auto-techsupport global  
SATATRATE LIMIT INTERVAL(sec) MAX TECHSUPPORT LIMIT(%) MAX CORE LIMIT(%) AVAILALE MEM THRESHOLD(%) MIN AVAILABLE MEM(kb) SINCE  
Enabled 180 10 5 10 200 2 days ago
```

show auto-techsupport-feature

- Usage:

```
show auto-techsupport-feature
```

- Example:

```
admin@sonic:~$ show auto-techsupport-feature
```

- Example:

```
admin@sonic:~$ show auto-techsupport-feature
```

| FEATURE NAME | STATE | RATE LIMIT INTERVAL (sec) | AVAILABLE MEM THRESHOLD (%) |
|--------------|---------|---------------------------|-----------------------------|
| bgp | enabled | 600 | 10 |
| database | enabled | 600 | 10 |
| dhcp_relay | enabled | 600 | 10 |
| lldp | enabled | 600 | 10 |
| swss | enabled | 800 | 10 |



show auto-techsupport history

- Usage:

```
show auto-techsupport history
```

- Example:

```
admin@sonic:~$ show auto-techsupport history
```

| TECHSUPPORT DUMP | DUMP TRIGGERED BY | EVENT TYPE | CORE |
|--|-------------------|------------|-------|
| ----- | ----- | ----- | ----- |
| sonic_dump_r-lionfish-16_20210901_221402 | bgp | core | |
| bgpcfgd.1630534439.55.core.gz sonic_dump_r-lionfish-16_20210901_203725 python3.1630528642.23.core.gz | snmp | core | |
| sonic_dump_r-lionfish-16_20210901_222408 | teamd | core | |
| python3.1630535045.34.core.gz | | | |
| sonic_dump_r-lionfish-16_20210901_222511 | N/A | memory | N/A |

2.2.19 Routing Stack

SONiC software is agnostic of the routing software that is being used in the device.

For example, users can use either Quagga or FRR routing stack as per their requirement. A separate shell (vtysh) is provided to configure such routing stacks. Once if users go to “vtysh”, they can use the routing stack specific commands as given in the following example.

- Example (Quagga Routing Stack):

```
admin@sonic:~$ vtysh
Hello, this is Quagga (version 0.99.24.1). Copyright 1996-2005
Kunihiro Ishiguro, et al.
sonic# show route-map
(This command displays the route-map that is configured for the routing protocol.)
ZEBRA:
route-map RM_SET_SRC, permit, sequence 10 Match
  clauses:
  Set clauses:
    src 10.12.0.102 Call
  clause: Action:
  Exit routemap
```

Refer the routing stack Quagga Command Reference or FRR Command Reference to know more about about the routing stack configuration.

2.2.20 Quagga BGP Show Commands

show ip bgp summary

This command displays the summary of all IPv4 bgp neighbors that are configured and the corresponding states.

- Usage:

```
show ip bgp summary
```

- Example:

```
admin@sonic:~$ show ip bgp summary
BGP router identifier 1.2.3.4, local AS number 65061 RIB entries 6124,
using 670 KiB of memory
Peers 2, using 143 KiB of memory
Neighbor      V      AS      MsgRcvd MsgSent TblVer  InQ    OutQ    Up/Down    State/PfxRcd
192.168.1.161 4       65501    88698   102781  0       0       08w5d14h    2
192.168.1.163 4       65502    88698   102780  0       0       08w5d14h    2
Total number of neighbors 2
Total number of neighbors 2
```



show ip bgp neighbors

This command displays all the details of IPv4 & IPv6 BGP neighbors when no optional argument is specified.

When the optional argument IPv4_address is specified, it displays the detailed neighbor information about that specific IPv4 neighbor.

Command has got additional optional arguments to display only the advertised routes, or the received routes, or all routes.

In order to get details for an IPv6 neighbor, use “show ipv6 bgp neighbor <ipv6_address>” command.

- Usage:

```
show ip bgp neighbors [<ipv4-address> [advertised-routes | received-routes | routes]]
```

- Example:

```
admin@sonic:~$ show ip bgp neighbors
BGP neighbor is 192.168.1.161, remote AS 65501, local AS 65061, external link
Description: DQS5K01T0
  BGP version 4, remote router ID 1.2.3.4
  BGP state = Established, up for 08w5d14h
  Last read 00:00:46, hold time is 180,
  keepalive interval is 60 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received Dynamic: received
    Route refresh: advertised and received(old & new) Address family IPv4 Unicast: advertised and received
    Graceful Restart Capabilty: advertised and received
    Remote Restart timer is 120 seconds Address families by peer:
      IPv4 Unicast(not preserved) Graceful restart informations:
    End-of-RIB send: IPv4 Unicast
    End-of-RIB received: IPv4 Unicast Message statistics:
    Inq depth is 0 Outq depth is 0

```

| | Sent | Rcvd |
|----------------|--------|-------|
| Opens: | 1 | 1 |
| Notifications: | 0 | 0 |
| Updates: | 14066 | 3 |
| Keepalive: | 88718 | 88698 |
| Route Refresh: | 0 | 0 |
| Capability: | 0 | 0 |
| Total: | 102785 | 8872 |

```

  Minimum time between advertisement runs is 30 seconds
  For address family: IPv4 Unicast
    Community attribute sent to this neighbor(both) 2 accepted prefixes
    Connections established 1; dropped 0 Last reset never
  Local host: 192.168.1.160, Local port: 32961

  Foreign host: 192.168.1.161,
  Foreign port: 179
  Nexthop: 192.168.1.160
  Nexthop global: fe80::f60f:1bff:fe89:bc00
  Nexthop local: ::
  BGP connection: non shared network
  Read thread: on
  Write thread: off

```





Optionally, you can specify an IP address in order to display only that particular neighbor. In this mode, you can optionally specify whether you want to display all routes advertised to the specified neighbor, all routes received from the specified neighbor or all routes (received and accepted) from the specified neighbor.

- Examples:

```
admin@sonic:~$ show ip bgp neighbors 192.168.1.161
admin@sonic:~$ show ip bgp neighbors 192.168.1.161 advertised-routes
admin@sonic:~$ show ip bgp neighbors 192.168.1.161 received-routes
admin@sonic:~$ show ip bgp neighbors 192.168.1.161 routes
```

show ipv6 bgp summary

This command displays the summary of all IPv4 bgp neighbors that are configured and the corresponding states.

- Usage:

```
show ipv6 bgp summary
```

- Example:

```
admin@sonic:~$ show ipv6 bgp summary
BGP router identifier 10.1.0.32, local AS number 65100 RIB entries 12809, using 1401 KiB of memory
Peers 8, using 36 KiB of memory
Neighbor      V      AS MsgRcvd   MsgSent  TblVer   InQ    OutQ   UP/Down   State/PfxRcd
fc00::72      4 64600 12588     12591    0        0      0    06:51:17    6402
fc00::76      4 64600 12587      6190    0        0      0    06:51:28    6402
fc00::7a      4 64600 12587      9391    0        0      0    06:51:23    6402
fc00::7e      4 64600 12589     12592    0        0      0    06:51:25    6402
Total number of neighbors 4
```

show ipv6 bgp neighbors

This command displays all the details of one particular IPv6 Border Gateway Protocol (BGP) neighbor. Option is also available to display only the advertised routes, or the received routes, or all routes.

- Usage:

```
show ipv6 bgp neighbors <ipv6-address> (advertised-routes | received-routes | routes)
```

- Examples:

```
admin@sonic:~$ show ipv6 bgp neighbors fc00::72 advertised-routes
admin@sonic:~$ show ipv6 bgp neighbors fc00::72 received-routes
admin@sonic:~$ show ipv6 bgp neighbors fc00::72 routes
```



show route-map

This command displays the routing policy that takes precedence over the other route processes that are configured.

- Usage:

```
show route-map
```

- Example:

```
admin@sonic:~$ show route-map
ZEBRA:
route-map RM_SET_SRC, permit, sequence 10 Match
  clauses:
  Set clauses:
    src 10.12.0.102 Call
  clause: Action:
    Exit routemap
ZEBRA:
route-map RM_SET_SRC6, permit, sequence 10 Match
  clauses:
  Set clauses:
    src fc00:1::102 Call
  clause: Action:
    Exit routemap
BGP:
route-map FROM_BGP_SPEAKER_V4, permit, sequence 10 Match
  clauses:
  Set clauses: Call
  clause: Action:
    Exit routemap
BGP:
route-map TO_BGP_SPEAKER_V4, deny, sequence 10 Match
  clauses:
  Set clauses: Call
  clause: Action:
    Exit routemap
BGP:
route-map ISOLATE, permit, sequence 10
  Match clauses: Set
  clauses:
    as-path prepend 65000 Call
  clause:
  Action:
    Exit routemap
```




2.2.21 ZTP Configuration and Show Commands

This section explains all the Zero Touch Provisioning commands that are supported in SONiC.

2.2.21.1 ZTP show commands

show ztp status This command displays the current ZTP configuration of the switch. It also displays detailed information about current state of a ZTP session. It displays information related to all configuration sections as defined in the switch provisioning information discovered in a particular ZTP session.

- Usage:

```
show ztp status [--verbose]
```

- Example:

```
root@B1-SP1-7712:/home/admin# show ztp status
ZTP Admin Mode      : True
ZTP Service         : Inactive
ZTP Status          : SUCCESS
ZTP Source          : dhcp-opt67 (eth0)
Runtime             : 05m 31s

ZTP Server is not running

01 - configdb-json: SUCCESS
02 - connectivity-check: SUCCESS
```

Use the verbose option to display more detailed information.

```
root@B1-SP1-7712:/home/admin# show ztp status
--verbose Command: ztp status --verbose
=====
ZTP
=====
ZTP Admin Mode      : True
ZTP Service         : Inactive
ZTP Status          : SUCCESS
ZTP Source          : dhcp-opt67 (eth0)
Runtime             : 05m 31s
Timestamp           : 2019-09-11 19:12:16 UTC
ZTP JSON Version    : 1.0

ZTP Service is not running
```



01-configdb-json

```

Status      : SUCCESS
Runtime     : 02m 48s
Timestamp   : 2019-09-11 19:11:55 UTC
Exit Code   : 0
Ignore Result : False

```

02-connectivity-check

```

Status      : SUCCESS
Runtime     : 04s
Timestamp   : 2019-09-11 19:12:16 UTC
Exit Code   : 0
Ignore Result : False

```

• Description

- **ZTP Admin Mode** - Displays if the ZTP feature is administratively enabled or disabled. Possible values are True or False. This value is configurable using “config ztp enabled” and “config ztp disable” commands.
- **ZTP Service** - Displays the ZTP service status. The following are possible values this field can display:
 - *Active Discovery*: ZTP service is operational and is performing DHCP discovery to learn switch provisioning information
 - *Processing*: ZTP service has discovered switch provisioning information and is processing it
- **ZTP Status** - Displays the current state and result of ZTP session. The following are possible values this field can display:
 - *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.
- **ZTP Source** - Displays the DHCP option and then interface name from which switch provisioning information has been discovered.
- **Runtime** - Displays the time taken for ZTP process to complete from start to finish. For individual configuration sections it indicates the time taken to process the associated configuration section.
- **Timestamp** - Displays the date/time stamp when the status field has last changed.
- **ZTP JSON Version** - Version of ZTP JSON file used for describing switch provisioning information.
- **Status** - Displays the current state and result of a configuration section. The following are possible values this field can display:
 - *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.



- **Exit Code** - Displays the program exit code of the configuration section executed. Non-zero exit code indicates that the configuration section has failed to execute successfully.
- **Ignore Result** - If this value is True, the result of the corresponding configuration section is ignored and not used to evaluate the overall ZTP result.
- **Activity String** - In addition to above information an activity string is displayed indicating the current action being performed by the ZTP service and how much time it has been performing the mentioned activity. Below is an example.
 - (04m 12s) Discovering provisioning data

2.2.21.2 ZTP configuration commands

This sub-section explains the list of the configuration options available for ZTP.

config ztp enable

Use this command to enable ZTP administrative mode

- Usage:

```
config ztp enable
```

- Example:

```
root@sonic:/home/admin# config ztp enable Running
command: ztp enable
```

config ztp disable

Use this command to disable ZTP administrative mode. This command can also be used to abort a current ZTP session and load the factory default switch configuration.

- Usage:

```
config ztp disable [-y|--yes]
```

- Example:

```
root@sonic:/home/admin# config ztp disable
Active ZTP session will be stopped and disabled, continue? [y/N]: y Running command: ztp
disable -y
```

config ztp run

Use this command to manually restart a new ZTP session. This command deletes the existing */etc/sonic/config_db.json* file and starts ZTP service. It also erases the previous ZTP session data. ZTP configuration is loaded on to the switch and ZTP discovery is performed.

- Usage:

```
config ztp run [-y|--yes]
```

- Example:

```
root@sonic:/home/admin# config ztp run
ZTP will be restarted. You may lose switch data and connectivity, continue? [y/N]: y
Running command: ztp run -y
```



2.3 SFP Utilities Commands

This sub-section explains the list of commands available for SFP utilities feature.

2.3.1 SFP Utilities show commands

- Show SFP EEPROM hex dump

```
admin@sonic:~$ sfputil show eeprom-hexdump --help Usage:
```

```
sfputil show eeprom-hexdump [OPTIONS]
```

Display EEPROM hexdump of SFP transceiver(s)

Options:

| | |
|--------------------------|---|
| -p, --port <port_name> | Display SFP EEPROM hexdump for port <port_name> Display |
| -n, --page <page_number> | SFP EEPROM hexdump for <page_number_in_hex> |
| | Show this message and exit. |
| --help | |

```
admin@sonic:~$ sfputil show eeprom-hexdump --port Ethernet0 --page 0
```

2.4 SFP Utilities read command

- Read SFP EEPROM data

```
admin@sonic:~$ sfputil read-eeprom --help
```

```
Usage: sfputil read-eeprom [OPTIONS]
```

Read SFP EEPROM data

Options:

| | |
|--------------------------------|--|
| -p, --port <logical_port_name> | Logical port name [required] |
| -n, --page <page> | EEPROM page number in hex [required] |
| -o, --offset <offset> | EEPROM offset within the page [required] |
| -s, --size <size> | Size of byte to be read [required] |
| --no-format | Display non formatted data Wire |
| --wire-addr TEXT | address of sff8472 |
| --help | Show this message and exit. |

```
admin@sonic:~$ sfputil read-eeprom -p Ethernet0 -n 0 -o 100 -s 2
```

```
admin@sonic:~$ sfputil read-eeprom --port Ethernet0 --page 0 --offset 0 --size 32
```

```
admin@sonic:~$ sfputil read-eeprom --port Ethernet0 --page 0 --offset 100 --size 2 --no-format
```

2.5 SFP Utilities write command

- Write SFP EEPROM data

```
admin@sonic:~$ sfputil write-eeprom --help
```

```
Usage: sfputil write-eeprom [OPTIONS]
```

Write SFP EEPROM data

Options:

| | |
|--------------------------------|--|
| -p, --port <logical_port_name> | Logical port name [required] |
| -n, --page <page> | EEPROM page number in hex [required] |
| -o, --offset <offset> | EEPROM offset within the page [required] |
| -d, --data <data> | Hex string EEPROM data [required] |
| --wire-addr TEXT | Wire address of sff8472 |
| --verify | Verify the data by reading back |
| --help | Show this message and exit. |



- Write success

```
admin@sonic:~$ sfputil write-eeprom -p Ethernet0 -n 0 -o 100 -d 4a44
admin@sonic:~$ sfputil write-eeprom --port Etherent0 --page 0 --offset 100 --data 0000 --verify
```

- Write fail

```
admin@sonic:~$ sfputil write-eeprom -p Etherent0 -n 0 -o 100 -d 4a44 --verify
Error: Write data failed! Write: 4a44, read: 0000.
```

2.6 Static DNS Commands

This sub-section explains the list of the configuration options available for static DNS feature.

2.6.1 Static DNS config command

- Add static DNS nameserver entry

```
admin@sonic:~$ config dns nameserver add -h
Usage: config dns nameserver add [OPTIONS] <ip_address>
```

Add static DNS nameserver entry

Options:

-?, -h, --help Show this message and exit.

- Delete static DNS nameserver entry

```
admin@sonic:~$ config dns nameserver del -h
Usage: config dns nameserver del [OPTIONS] <ip_address>
```

Delete static DNS nameserver entry

Options:

-h, -?, --help Show this message and exit.

2.6.2 Static DNS show command

- Show static DNS configuration

```
admin@sonic:~$ show dns nameserver -h Usage:
show dns nameserver [OPTIONS]
```

Show static DNS configuration

Options:

-h, -?, --help Show this message and exit.

```
admin@sonic:~$ show dns nameserver
```

Nameserver

1.1.1.1

8.8.8.8



2.7 Wake-on-LAN Commands

2.7.1 Send Wake-on-LAN Magic Packet command

The `wol` command is used to send magic packet to target device.

2.7.1.1 Usage

```
wol <interface> <target_mac> [-b] [-p password] [-c count] [-i interval] [-v]
```

- **interface** : SONiC interface name.
- **target_mac** : a list of target devices' MAC address, separated by comma.
- **-b** : Use broadcast MAC address instead of target device's MAC address as **Destination MAC Address in Ethernet Frame Header**.
- **-p password** : An optional 4 or 6 byte password, in ethernet hex format or quad-dotted decimal[^3].
- **-c count** : For each target MAC address, the `count` of magic packets to send. `count` must between 1 and 5. Default value is 1. This param must use with `-i`.
- **-i interval** : Wait `interval` milliseconds between sending each magic packet. `interval` must between 0 and 2000. Default value is 0. This param must use with `-c`.
- **-v** : Verbose output.

2.7.1.2 Example

```
admin@sonic:~$ wol Ethernet10 00:11:22:33:44:55
admin@sonic:~$ wol Ethernet10 00:11:22:33:44:55 -b
admin@sonic:~$ wol Vlan1000 00:11:22:33:44:55,11:33:55:77:99:bb -p 00:22:44:66:88:aa
admin@sonic:~$ wol Vlan1000 00:11:22:33:44:55,11:33:55:77:99:bb -p 192.168.1.1 -c 3 -i 2000 -v
Sending 3 magic packet to 00:11:22:33:44:55 via interface Vlan1000
1st magic packet sent to 00:11:22:33:44:55
2nd magic packet sent to 00:11:22:33:44:55
3rd magic packet sent to 00:11:22:33:44:55
Sending 3 magic packet to 11:33:55:77:99:bb via interface Vlan1000
1st magic packet sent to 11:33:55:77:99:bb
2nd magic packet sent to 11:33:55:77:99:bb
3rd magic packet sent to 11:33:55:77:99:bb
```

For the 4th example, it specifies 2 target MAC addresses and `count` is 3. So it'll send 6 magic packets in total.



2.8 Banner Commands

This sub-section explains the list of the configuration options available for Banner feature.

2.8.1 Banner config commands

- Set banner feature state

```
admin@sonic:~$ config banner state <enabled|disabled>
```

```
Usage: config config banner state <enabled|disabled>
```

Set banner feature state

Options:

-?, -h, --help Show this message and exit.

- Set login message

```
admin@sonic:~$ config banner login <message>
```

```
Usage: config banner login <message>
```

Set login message

Options:

-?, -h, --help Show this message and exit.

- Set logout message

```
admin@sonic:~$ config banner logout <message>
```

```
Usage: config banner logout <message>
```

Set logout message

Options:

-?, -h, --help Show this message and exit.

- Set message of the day

```
admin@sonic:~$ config banner motd <message>
```

```
Usage: config banner motd <message>
```

Set message of the day

Options:

-?, -h, --help Show this message and exit.



2.8.2 Banner show command

- how banner messages

```
admin@sonic:~$ show banner
Usage: show banner
```

Show banner messages

Options:

-h, -?, --help Show this message and exit.

```
admin@sonic:~$ show banner
state      login      motd      logout
enabled    Login     Your are on Message

  _____
 /  _  _  \
_ \|			
	_)		
_ \|			
_	\_	_	_

-- Software for Open Networking in the Cloud --

Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:      https://sonic-net.github.io/SONiC/
```

2.9 Memory Statistics Commands

2.9.1 Overview

These commands allow users to enable/disable memory statistics monitoring, configure data collection intervals, adjust data retention periods, view memory statistics, and check the current configuration. Memory statistics can help administrators monitor and analyze system memory usage over time.

Common Use Cases

- Monitor system memory trends over time.
- Track memory usage patterns during peak time.
- Plan system capacity based on historical memory data.

2.9.2 Memory Statistics Config Commands

2.9.2.1 Enable/Disable Memory Statistics Monitoring

To enable or disable the memory statistics monitoring feature:

```
admin@sonic:~$ config memory-stats enable/disable
```

This will **enable/disable** memory statistics monitoring.

By default, this feature is **disabled**.

Examples:

- To enable memory statistics monitoring:

```
admin@sonic:~$ config memory-stats enable Memory statistics monitoring enabled.
```
- To disable memory statistics monitoring:

```
admin@sonic:~$ config memory-stats disable Memory statistics monitoring disabled.
```




2.9.2.2 Set the Frequency of Memory Data Collection

To configure the interval for memory data collection (specified in minutes):

```
admin@sonic:~$ config memory-stats sampling-interval <interval>
```

- <interval> is the sampling interval in minutes.
- The default sampling interval is **5 minutes**.

Example:

- To set the sampling interval to 10 minutes:

```
admin@sonic:~$ config memory-stats sampling-interval 10 Sampling interval set to 10 minutes.
```

2.9.2.3 Adjust the Data Retention Period

To set how long the memory data should be retained (specified in days):

```
admin@sonic:~$ config memory-stats retention-period <period>
```

- <period> is the retention period in days.
- The default retention period is **15 days**.

Example:

- To set the retention period to 30 days:

```
admin@sonic:~$ config memory-stats retention-period 30 Retention period set to 30 days.
```

2.9.3 Memory Statistics Show Commands

2.9.3.1 View Memory Usage Statistics

To display memory usage statistics, use the following command with optional parameters for time range and specific metrics:

```
admin@sonic:~$ show memory-stats [--from <date-time>] [--to <date-time>] [--select <metric>]
```

Command Options:

- **show memory-stats** : Display basic memory usage statistics.
- **--from <date-time>** : Display memory statistics from the specified start date-time.
- **--to <date-time>** : Display memory statistics up to the specified end date-time.
- **--select <metric>** : Display specific memory statistics, such as total memory.

Time Format for Statistics Retrieval: The time format for **--from** and **--to** options includes:

- **Relative time formats:**
 - 'X days ago', 'X hours ago', 'X minutes ago'
 - 'yesterday', 'today'
- **Specific times and dates:**
 - 'now'
 - 'July 23', 'July 23, 2024', '2 November 2024' ▸ '7/24', '1/2'
- **Time expressions:**
 - '2 am', '3:15 pm'
 - 'Aug 01 06:43:40', 'July 1 3:00:00'
- **Named months:**
 - 'jan', 'feb', 'march', 'september', etc.
 - Full month names: 'January', 'February', 'March', etc.
- **ISO 8601 format:**
 - '2024-07-01T15:00:00'



2.9.3.2 Default Historical Memory Statistics

To view the historical memory statistics:

```
admin@sonic:~$ show memory-stats
```

2.9.3.3 Historical Memory Statistics for Last 10 Days

To view memory statistics for the last 10 days:

```
admin@sonic:~$ show memory-stats --from '10 days ago' --to 'now'
```

2.9.3.4 Historical Memory Statistics for Last 100 Minutes

```
admin@sonic:~$ show memory-stats --from '100 minutes ago' --to 'now'
```

2.9.3.5 Historical Memory Statistics for Last 3 Hours

```
admin@sonic:~$ show memory-stats --from '3 hours ago' --to 'now'
```

2.9.3.6 Historical Memory Statistics for Specific Metric (Used Memory)

```
admin@sonic:~$ show memory-stats --from '100 minutes ago' --to 'now' --select 'used_memory'
```

2.9.3.7 View Memory Statistics Configuration

To display the current configuration parameters such as data collection frequency, retention period, and enable/disable status, use the following command:

```
admin@sonic:~$ show memory-stats-config
```

Example:

```
admin@sonic:~$ show memory-stats-config
```

Mmory Static Configuration:

```
-----
Enabled           : false
Sampling Interval  : 5
Retention Period   : 15
```