



Extreme ONE OS Switching v22.2.0.0 Management Configuration Guide

Comprehensive Instructions for Configuration and
Management

9039333-00 Rev AA
July 2025



Copyright © 2025 Extreme Networks, Inc. All rights reserved.

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: <https://www.extremenetworks.com/about-extreme-networks/company/legal/trademarks>

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses.

End-user license agreements and open source declarations can be found at: <https://www.extremenetworks.com/support/policies/open-source-declaration/>



Table of Contents

Abstract.....	viii
Preface.....	ix
Text Conventions.....	ix
Documentation and Training.....	x
Open Source Declarations.....	xi
Training.....	xi
Help and Support.....	xi
Subscribe to Product Announcements.....	xii
Send Feedback.....	xii
About This Document.....	13
What's New in this Document	13
Regarding Ethernet Interfaces and Chassis Devices.....	13
User Management.....	13
Configuration Fundamentals.....	15
Configuration Files.....	15
Default Configuration Files.....	15
Running Configuration Files.....	16
Auto-Persistence of Configuration Data.....	16
Displaying Configurations.....	16
Backing up Configurations.....	17
Configuration File Restoration.....	18
Managing Flash Files.....	19
Copying Support Save Files	20
Session Connection.....	22
Telnet.....	22
SSH.....	24
Configuring Terminal Session Parameters.....	27
Configuring Login Banner.....	27
Configuring MOTD Banner.....	28
System Logging Configuration.....	29
Ethernet Management Interfaces.....	30
Configuring Static Management Port.....	30
Displaying the Management Interface.....	31
Configuring an IPv6 Address on the ONE OS Platform.....	33
Port Management.....	33
Extreme ONE OS 100G Ports.....	33
Link Fault Signaling.....	34
Dual OOB Redundant Management Port Configuration.....	35
Interface Ethernet Ports.....	36
Displaying Device Interfaces.....	36
System Clock.....	37

Setting the Clock.....	37
Management VRFs.....	37
VRF Reachability.....	38
MAC Address Aging.....	40
BGP Protocol Event Monitoring and Notification.....	40
Supported Functionalities.....	40
BGP Enterprise and Standard MIB Notifications.....	41
gNMI Notifications.....	42
RAS Logs.....	42
CLI Commands.....	43
Support for OpenConfig Telemetry	44
OpenConfig-BGP Yang Module	45
OpenConfig-Platform Yang Module	46
Securing OpenConfig Telemetry Connections	46
Zero Touch Provisioning (ZTP) over DHCP.....	46
ZTP Configuration.....	46
Config and DRC (Drift & Reconcile) Tracking.....	50
Key Features and Capabilities	51
Persistence Handling in CDB vs SDB Reload.....	51
ONE OS and Linux Shell Interoperability.....	53
Overview	53
Limitations	53
Accessing the Linux Shell from ONE OS.....	53
Escalating Linux Permissions.....	54
Saving and Appending Show Command Output to a File.....	54
Logs of Linux Shell Activities.....	55
Firmware Fullinstall Support.....	55
Key Features.....	55
Event Log Messages.....	55
Important Extreme ONE OS Configuration and Certificate Changes.....	55
CLI Commands.....	56
TPVM IAH Extension.....	60
About Integrated Appliance Hosting (IAH).....	60
Configure TPVM.....	61
Config CLI.....	62
ONE OS IAH Equivalence.....	62
Exec CLI or gNOI.....	68
ONE OS IAH Equivalence.....	69
Show or gNOI CLI.....	70
ONE OS IAH Equivalence.....	70
RASlog Errors or Logs.....	72
ONE OS IAH Equivalence.....	72
Network Time Protocol (NTP).....	74
Network Time Protocol overview.....	74
Date and Time Settings.....	74
Time Zone Settings.....	74
Network Time Protocol Server Overview.....	75
Network Time Protocol Client Overview.....	75

Network Time Protocol Associations.....	76
Configuring NTP.....	76
Displaying the NTP Server Status.....	77
NTP Server Status When an NTP Server is Not Configured.....	77
NTP Server Status When an NTP Server is Configured.....	77
SNMP.....	78
SNMP Overview.....	78
Basic SNMP Operation.....	80
SNMP Community Strings.....	80
SNMP Users.....	80
SNMP Server Hosts.....	81
SNMP Source Interface.....	81
Configuring SNMPv2.....	81
Configuring SNMPv3.....	82
Supported MIBs.....	83
LLDP.....	85
LLDP Overview.....	85
Layer 2 Topology Mapping.....	85
LLDP Configuration Guidelines and Restrictions.....	93
Configuring and Managing LLDP.....	94
Understanding the Default LLDP.....	94
Disabling LLDP Globally.....	94
Configuring LLDP Global Parameters.....	94
Displaying LLDP Information.....	95
Clearing LLDP-Related Information.....	98
gNMI and gNOI.....	99
gNMI and gNOI Overview.....	99
Troubleshooting Information.....	99
gNMI Overview.....	100
gNOI Overview.....	101
gRPC Network Management Interface (gNMI) Service.....	101
Capabilities RPC.....	102
Get RPC.....	102
Set RPC.....	102
Subscribe RPC.....	103
gRPC Network Operational Interface (gNOI) Service.....	104
gNOI vs gNMI.....	104
Configure gRPC Server.....	104
Data Model and Northbound Interface.....	105
Storage of List Key Values.....	106
Inband Management Support.....	106
CLI Commands.....	106
Config Commands.....	106
Exec Commands.....	106
Event Log Messages.....	107
gNMI Authentication and Encryption.....	107
Credentials and Authentication.....	107
Importing gNMI Client CA Root Certificate	107

Security.....	108
Encryption.....	108
Authentication.....	108
BMC Configuration.....	109
Increase BMC Security	109
Intelligent Platform Management Interface	109
Baseboard Management Controller	110
Securing BMC	110
Change BMC User Password	110
Enable the BMC Management Interface	110
Configure BMC Management Interface IP Address.....	111
Reset BMC Configuration to Factory Defaults	113
Port Mirroring.....	114
Mirroring Overview.....	114
Types of Mirroring Supported.....	114
Limitations.....	114
Configuration Guidelines.....	114
Configuration Validations for Mirroring.....	115
Single Destination Interface per Mirror Session.....	115
No Port-Channel Members as Destination Interfaces.....	115
No Port-Channel Members as Source Interfaces.....	115
YANG Modules for Mirroring.....	115
Extreme Mirror YANG Module.....	115
OpenConfig ACL YANG Augmentation.....	116
OpenConfig Interfaces YANG Augmentation.....	116
CLI Commands for Mirroring.....	117
Mirror Destination Creation.....	117
Port-Based Mirroring.....	117
Use Case Scenarios and Configuration Examples for Port Mirroring.....	117
Local SPAN Port-Based Mirroring.....	118
Maintenance Mode.....	119
Maintenance Mode Overview.....	119
Key Benefits.....	119
Configuration Options.....	119
Enabling or Disabling Maintenance Mode.....	119
Maintenance Mode Enable Sequence.....	120
Maintenance Mode Disable Sequence.....	120
Maintenance Mode On Reload.....	120
MLAG and BGP Module Behavior.....	120
MLAG Bring-up Delay.....	120
BGP Module Behavior.....	121
Restart Handling.....	121
CLI Commands for Maintenance Mode.....	121
Operational Commands.....	121
Configuration Commands.....	122
MLAG Commands.....	122
Show Tech Additions.....	123
RASLog and SNMP Traps for Maintenance Mode.....	124

RASLog Messages.....	124
SNMP MIBs and Traps.....	124
SNMP trap daemon output.....	124
Event Log Messages.....	126
Enter Maintenance Mode Before Performing Device Maintenance.....	126
Rebooting into Maintenance Mode	127



Abstract

This management configuration guide for Extreme ONE OS Switching version 22.2.0.0 provides authoritative CLI-based procedures for configuring and operating Extreme Networks switching platforms. It covers persistent and runtime configuration management, including auto-persistence mechanisms, backup and restoration workflows, and flash file system operations. The guide details interface provisioning, including Ethernet, management, and loopback interfaces, with support for IPv4/IPv6 addressing, VRF-aware routing, and dual out-of-band management redundancy. It includes comprehensive instructions for configuring BGP with support for L2VPN EVPN, OpenConfig telemetry via gNMI/gNOI, and SNMPv2/v3 with MIB and trap configuration. Security features include SSH/Telnet access control, RBAC enforcement, and certificate-based authentication for gRPC services. Zero Touch Provisioning (ZTP) is supported via DHCP options 66/67/43, enabling automated firmware and configuration deployment. The guide introduces Integrated Appliance Hosting (IAH) for deploying TPVM virtual machines using KVM and libvirt, with full CLI and gNMI configuration paths. Additional modules cover system logging, LLDP with custom TLVs, MAC address aging, and configuration drift detection via DRC tracking. Designed for experienced network engineers and administrators, the guide emphasizes deterministic behavior, secure automation, and operational introspection across Extreme ONE OS deployments.



Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as Extreme Networks switches or routers, the product is referred to as *the switch* or *the router*.

Table 1: Notes and warnings






Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
screen displays	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words <i>enter</i> and <i>type</i>	When you see the word <i>enter</i> in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says <i>type</i> .
Key names	Key names are written in boldface, for example Ctrl or Esc . If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press Ctrl+Alt+Del
<i>Words in italicized type</i>	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
<i>italic</i> text	Italic text indicates variable content.
[]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x y z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member[member...]</i> .
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

[Current Product Documentation](#)

[Release Notes](#)

[Hardware and Software Compatibility](#) for Extreme Networks products

[Extreme Optics Compatibility](#)

[Other Resources](#) such as articles, white papers, and case studies

Open Source Declarations

Some software files have been licensed under certain open source licenses. Information is available on the [Open Source Declaration](#) page.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit the [Extreme Networks Training](#) page.

Help and Support

If you require assistance, contact Extreme Networks using one of the following methods:

[Extreme Portal](#)

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

[The Hub](#)

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

[Call GTAC](#)

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2800. For the support phone number in your country, visit www.extremenetworks.com/support/contact.

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Product Announcements

You can subscribe to email notifications for product and software release announcements, Field Notices, and Vulnerability Notices.

1. Go to [The Hub](#).
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

You can modify your product selections or unsubscribe at any time.

Send Feedback

The User Enablement team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, email us at Product-Documentation@extremenetworks.com.

Provide as much detail as possible including the publication title, topic heading, and page number (if applicable), along with your comments and suggestions for improvement.



About This Document

[What's New in this Document](#) on page 13

[Regarding Ethernet Interfaces and Chassis Devices](#) on page 13

[User Management](#) on page 13

What's New in this Document

This document is new for the Extreme ONE OS Switching v22.2.2.0 software release.

For additional information, refer to the *Extreme ONE OS Switching Release Notes*.

Regarding Ethernet Interfaces and Chassis Devices

The current version does not support any multi-slot (chassis) devices.

However, the Ethernet interface configuration and output *slot/port* examples in this document may appear as either 0/x or n/x, where "n" and "x" are integers greater than 0.

For all currently supported devices, specify 0 for the slot number.

User Management

The device is pre-configured with a default username 'admin'. Upon first login, you will be prompted to update the default password. For more information, see [Force Password Change At First Login](#).

During the first login via CLI, users will be prompted to change both the admin password and the GRUB password. Users can choose to either set a different password for GRUB or use the same admin password for GRUB (by pressing Enter). After updating the passwords, the system will confirm that the admin and GRUB passwords have been updated successfully.

To reconfigure the GRUB password, use the CLI command: `config terminal > system > grub > username <name> password <password>`. After a factory reset, both admin and GRUB passwords will be reset to <default-password> and must be changed by the user. When entering the ONIE prompt during reload, users will need to enter the GRUB login credentials (username: root, password: user-set password).

The new password must meet the following criteria:

- Minimum of 8 characters
- At least one uppercase letter
- At least one number
- At least one special character (for example, aRng#456)
- Must include at least one lowercase character



Note

- Running the **copy default-config running-config** command will reset the password, prompting you to set a new one.
- SSH is the only protocol enabled by default in the system.



Configuration Fundamentals

[Configuration Files](#) on page 15
[Session Connection](#) on page 22
[System Logging Configuration](#) on page 29
[Ethernet Management Interfaces](#) on page 30
[Configuring an IPv6 Address on the ONE OS Platform](#) on page 33
[Port Management](#) on page 33
[Interface Ethernet Ports](#) on page 36
[System Clock](#) on page 37
[Management VRFs](#) on page 37
[MAC Address Aging](#) on page 40
[BGP Protocol Event Monitoring and Notification](#) on page 40
[Support for OpenConfig Telemetry](#) on page 44
[Zero Touch Provisioning \(ZTP\) over DHCP](#) on page 46
[Config and DRC \(Drift & Reconcile\) Tracking](#) on page 50

Configuration Files

When you boot up a device for the first time, the default configuration is the running configuration. As you configure the device, the changes are written to the running configuration. Changes to the running configuration are auto persisted across device reboots.

Default Configuration Files

Default configuration files are part of the firmware package for the device and are automatically applied to the running configuration under the following conditions:

- When the device boots up for the first time and no customized configuration is available.
- When you restore the default configuration.

You cannot remove, rename, or change the default configuration.

Running Configuration Files

The configuration currently effective on the device is referred to as the running configuration. Any configuration change you make while the device is online are made to the running configuration.

The running configuration file name is running-config.

Auto-Persistence of Configuration Data

All configuration changes performed on the device are automatically persisted by default.

Configurations are saved to a persistent configuration data store. After a device reboots, or, is brought up, configurations are restored from this data store. If this data store becomes unusable for any reason, configurations are then replayed from the default-config file.

Displaying Configurations

The following examples illustrate how to display the running configuration.

Displaying the running configuration

To display the contents of the running configuration, enter the **show running-config** command in the privileged EXEC mode. Following output is truncated.

```
device# show running-config
! Firmware: 10.2.0.0, Application(s): 22.2.0.0
! Date and Time: Tue, 2025-05-05 06:35:20 EST
system
  clock timezone America/New_York
  hostname device
  ntp
    server xxx.0.x.0
    enable
  !
  aaa
    authentication
      admin-user admin password-hashed $6$PlcHoBtYr2gs/WHJ$.mCwD6S23B5/JJpED7mGV
      54JdO4WsyCi6xs.wYeRGtYfKWYucKWtwj/PyKvm.P0DMIirxaFo6Zj0bzgJ5JbSv0
    !
  !
  grpc-server DEFAULT
    certificate-id ssl-reserved-generated
    port 443
    enable
  !
  !
  ssh-server DEFAULT
    vrf mgmt-vrf
    enable
  !
  logging
    console severity critical
  !
  !
bfd
```



```
profile default
  interval 300 min-rx 300 multiplier 3
!
profile mlag
  interval 300 min-rx 300 multiplier 3
!
!
overlay-networks
  default-nve base
  default-vni-offset 0
  nvo base
    member nve base
    member vni-domain base
    split-horizon base
  !
  nve base
  vni-domain base
!
interface ethernet 0/1
  no shutdown
!
interface ethernet 0/2
  no shutdown
!
interface ethernet 0/7
  shutdown
!
interface ethernet 0/9
  shutdown
!
interface ethernet 0/17
  no shutdown
  subinterface vlan 500
    no shutdown
  !
!
interface ethernet 0/21
  no shutdown
  subinterface vlan 500
    no shutdown
  !
!
interface management 0
  ipv4 address 1xx.0.x.x/24
  ipv6 address dhcp
  no shutdown
!
vrf mgmt-vrf
  member management 0
  static-route 0.x.0.x/x 192.0.x.x
vrf default-vrf
  control-plane
device#
```

Backing up Configurations

Always keep a backup copy of your configuration files, so you can restore the configuration in the event the configuration is lost or you make unintentional changes.

The following recommendations apply:

- Upload the configuration backup copies to an external host or to an attached Extreme-branded USB device.
- Avoid copying configuration files from one device to another. Instead restore the device configuration files from the backup copy.

Copying a Configuration File to an External Host

In the following example, the running configuration is copied to a file on a remote server using SCP:

```
device# copy running-config file scp://root:test@123@192.0.2.0/root/temp/
runningConfig1.conf
device#
```

Backing up the Running Configuration to a USB Device

When you make a backup copy of a configuration file on an attached USB device, specify the USB and the destination file name on the USB device. You do not need to specify the target directory. The file is automatically recognized as a configuration file and stored in the default configuration directory.

1. Enable the USB device.

```
device# usb enable
device#
```

2. Copy running-config to the destination USB.

```
device# copy running-config file usb://temp.cfg
```

Configuration File Restoration

Restoring a configuration involves overwriting a given configuration file on the device by downloading an archived backup copy from an external host or an attached device.

All interfaces remain online. The following parameters are unaffected:

- Interface management IP address
- Software feature licenses installed on the device

The following is an example output of restoring a configuration:

```
device# copy file scp://user@192.x.x.x/home/user/test.cfg running-config
```

Restoring the Default Configuration

To restore the default configuration, perform the following procedure in EXEC mode.

1. Enter the **copy default-config running-config** command to overwrite the running configuration with the default configuration from a saved configuration file.

```
device# copy default-config running-config
This operation will modify your running configuration.
```

2. Confirm that you want to make the change by entering <user input> when prompted.

```
WARN: system will be rebooted to have configuration changes to take effect!
Do you want to continue? [y/n]: n
Copy operation aborted by user
```

Managing Flash Files

The Extreme device provides a set of tools for removing, and displaying files you create in the device flash memory. You can use the display commands with any file, including the system configuration files. The **rm** command only applies to copies of configuration files you create in the flash memory. You cannot delete any of the system configuration files.

Listing Contents of the Flash Memory

To list the contents of the flash memory, enter the **ls** command in EXEC mode.

```
device# ls
config-file  :
pcap-file   :
-rw-r--r--  0  2025-02-25 19:32:33  README.md
tech-support :
-rw-----  6148      2025-02-13 09:09:03  TS-device-250213T0843.log
-rw-r--r--  11488697 2025-02-13 09:09:03  TS-device-250213T0843.tar.gz
firmware    :
coredumps   :
iah         :
drwxr-xr-x  4096    2025-02-24 23:04:32  efaboot
```

View the Content of a File in Flash Memory

Use the **cat** command to view file contents.

```
device# cat disk://config-file/testConfig.conf
! Firmware: 10.2.0.0, Application(s): 22.2.0.0
! Date and Time: Mon, 2025-03-10 00:54:36 EDT
system
  clock timezone  America/New_York
  hostname device
  ntp
    server 192.x.x.x
    enable
  !
  aaa
    authentication
      admin-user admin password-hashed $6$IFcKQDE68LWEjxCP$JZgqUWvQLv3oEBxo4JMBi
//6/Po5hkbBMNqnSlyPDORUKOELDdGXsVQldRJ.5EksCCnx9dPwTrT.zR1FrCECX1
    !
  !
  grpc-server DEFAULT
    certificate-id ssl-reserved-generated
    mutual-tls
    port 9340
    vrf vrfnew1
  !
```

```
grpc-server gnminewl
certificate-id test_cert_id
```

**Note**

To display the contents of the running configuration, use the `show running-config` command.

Deleting File from the Flash Memory

To delete a file from the flash memory, enter the `rm` command with the file name in EXEC mode.

```
device# rm disk://config-file/testConfig.cfg
Warning: File testConfig.cfg will be deleted (from disk).
Do you want to continue? [y/n]: y
```

**Note**

You cannot delete a system configuration file in flash memory.

Copying Support Save Files

The following is an example CLI output for system techsupport generation using the **system tech-support** command:

```
device# system tech-support
Generating show commands output
Executing base-svc host commands
Executing commands for service chassis-mgr
Executing commands for service state-db
Executing commands for service bgp
Executing commands for service mftm
Executing commands for service uftm
Executing commands for service mlag
Executing commands for service arpnd
Executing commands for service fwd-hal
Executing commands for service bfd
Executing commands for service iah
Executing commands for service telegraf
Executing commands for service classifiers
Executing commands for service chassis-mgr
Executing commands for service state-db
Executing commands for service interface-mgr
Archiving /var/log/tierra/vpp-sr/vpp
Archiving /etc/libvirt
Archiving /mnt/onl/iah/logs
Archiving /etc/extreme
Archiving /var/log/messages
Archiving /var/log/messages.1
Archiving /var/log/messages.1.gz
Archiving /var/log/messages.2
Archiving /var/log/messages.2.gz
Archiving /var/log/messages.3
Archiving /var/log/messages.3.gz
Archiving /var/log/messages.4
Archiving /var/log/messages.4.gz
Archiving /var/log/auth.log
Archiving /var/log/debug
Archiving /var/log/debug.1
Archiving /var/log/debug.1.gz
```

```

Archiving /var/log/debug.2
Archiving /var/log/debug.2.gz
Archiving /var/log/debug.3
Archiving /var/log/debug.3.gz
Archiving /var/log/debug.4
Archiving /var/log/debug.4.gz
Archiving /var/log/containers
Archiving /var/log/tierra
Archiving /var/log/log_file_journalctl
Archiving /var/log/daemon.log
Archiving /var/log/bmclog.gz
Archiving /var/log/consolelog
Archiving /tmp/tierra
Archiving /mnt/onl/usrdata/coredumps
Archiving /mnt/onl/usrdata/oom
Archiving /mnt/onl/usrdata/pprof
Archiving /var/data/disk/techsupport/ShowOutput.txt
Archiving /var/data/disk/techsupport/.prevJournalLog
Archiving /var/ms-commands
Archiving /mnt/onl/config/auditlog
Archiving /mnt/onl/config/db
Archiving /mnt/onl/config/allHwTables.txt
Archiving /var/data/monitors/k3s_events
Archiving /var/data/monitors/pods_resources_usage.log.gz
Archiving /var/data/monitors/system_resources_usage.log.gz
Archiving /etc/snmp
Archiving /var/log/snmpd.log
Archiving /var/log/ethphyd.bootlog
Archiving /etc/passwd
Archiving /etc/shadow
Archiving /etc/group
Archiving /etc/ssh
Archiving /etc/tierra
Archiving /etc/ldap.conf
Archiving /etc/ldap.map
Archiving /etc/tacacs.conf
Archiving /etc/radius.conf
Archiving /etc/pam.d
Archiving /var/data/cert-mgmt/certs
Archiving /var/data/cert-mgmt/app-cert
Archiving /var/data/cert-mgmt/ca-trust
Archiving /lib/systemd/system/xinetd*.service
Archiving /lib/systemd/system/sshd*.service
Archiving /var/crash
Archiving /var/log/tierra-svc-stdoutlogs
Archiving /var/data/ztp
Cleaning up temporary files
TechSupport completed and file TS-kvm-x86-64-250307T1956.tar.gz generated

Done
device#

device# ls
config-file :
pcap-file :
-rw-r--r-- 0 2025-03-07 04:32:14 README.md
tech-support :
-rw----- 6170 2025-03-07 09:20:13 TS-kvm-x86-64-250307T1956.log
-rw-r--r-- 13823118 2025-03-07 09:20:13 TS-kvm-x86-64-250307T1956.tar.gz
firmware :
```

```

coredumps :
iah :

device# copy file disk://tech-support/TS-kvm-x86-64-250307T1956.tar.gz file scp://
user:password@192.x.x.x:/home/user
copy file to a file is successfull

```

A new directory with the format `<TS>-<HOSTNAME>-<YYMMDDTHHMM>.log` file is automatically saved in the internal techsupport directory within the switch or USB device. The location of this sub-directory will depend on the **copy** command parameters.

```

tech-support :
-rw----- 7378      2025-03-06 23:43:37 TS-8730-32d-250306T4006.log
-rw-r--r-- 15436035 2025-03-06 23:43:37 TS-8730-32d-250306T4006.tar.gz

```

You can transfer the techsupport file to a USB drive or an external server.

```

device# copy file disk://tech-support/TS-8730-32d-250306T4006.tar.gz file usb://
TS-8730-32d-250306T4006.tar.gz

```

You can transfer the tech support file to a remote server.

```

device# system tech-support scp://prashanth:123!#aBc@1.2.3.4/home/user/techsupport/TS/

```

Session Connection

You can connect to your device through a console session on the serial port, or through a Telnet or Secure Shell (SSH) connection to the management port belonging to either the mgmt-vrf, default-vrf, or a user-defined vrf. You can use any account login present in the local device database or on a configured authentication, authorization, and accounting (AAA) server for authentication. For initial setup procedures, use the pre-configured administrative account that is part of the default device configuration.

The device must be physically connected to the network. If the device network interface is not configured or the device has been disconnected from the network, use a console session on the serial port.

Refer to the appropriate hardware guide for information on connecting through the serial port and establishing an Ethernet connection for a console session.



Warning

If you try to create more than 32 CLI sessions (either SSH or TELNET sessions, or a combination of both sessions), a message will be displayed to close one of the existing sessions to proceed.

Telnet

Telnet allows access to management functions on a remote networking device. Unlike SSH, Telnet does not provide a secure, encrypted connection to the device.

The device supports a combined maximum (SSH, Telnet, and serial) of 32 CLI sessions. Both IPv4 and IPv6 addresses are supported.

The Telnet service is disabled by default on the device. When the Telnet server is disabled, existing inbound Telnet connections are terminated and access to the device by additional inbound connections is not allowed until the Telnet server is re-enabled. If you have admin privileges, you can disable and re-enable inbound Telnet connections from global configuration mode.

Connecting to an Extreme Device with Telnet

A Telnet session allows you to access a device remotely using port 23. However, it is not secure. If you need a secure connection, use SSH.

1. Establish a Telnet session to the Extreme device from a remote device.

```
device# telnet 192.x.x.x
```

The example establishes a Telnet session to the device with the IP address of 192.x.x.x

If the device is active and the Telnet service is enabled on it, a display similar to the following appears.

```
Trying 192.x.x.x...
Connected to 192.x.x.x.
Escape character is '^]'.
```

2. Once you have established the Telnet connection, you can log in normally.

```
login as: admin
Pre-authentication banner message from server:
This system is for authorized users only. All activity is logged and regularly checked
by systems personal. Individuals using this system without authority or in excess
of their authority are subject to having all their services revoked. Any illegal
activities conducted by user or attempts to take down this system or its services will
be reported to local law enforcement and said user will be punished to the full extent
of the law. Anyone using this system consents to these terms.
End of banner message from server
admin@192.x.x.x's password:
Last login: Mon Mar  3 07:17:02 2025 from 192.x.x.x
*****
System is ready for all commands
*****
device#
```

The default admin login name is admin. Extreme recommends that you change the default account password when you log in for the first time.

Enable or Disable Telnet Service

1. In EXEC mode, enter global configuration mode.

```
device# configure terminal
device(config)#
```

2. Disable Telnet service on the device.

```
device(config-system-telnet-server-DEFAULT)# no enable
device(config-system-telnet-server-DEFAULT)#
```

All Telnet sessions including any currently active sessions are immediately terminated, and cannot be re-established until the service is re-enabled.

3. Enable Telnet service on the device.

```
device(config-system-telnet-server-DEFAULT)# enable
device(config-system-telnet-server-DEFAULT)#
```

SSH

Secure Shell (SSH) allows secure access to management functions on a remote networking device. Unlike Telnet, which offers no security, SSH provides a secure, encrypted connection to the device.

SSH support is available in EXEC mode on all Extreme platforms. The device supports a combined maximum (SSH, Telnet, and serial) of 32 CLI sessions. Both IPv4 and IPv6 addresses are supported.

The SSH service is enabled by default on the device. When the SSH server is disabled, existing SSH connections will remain active until they reach their idle timeout. Additionally, new session access will be blocked until the SSH server is re-enabled. If you have admin privileges, you can disable and re-enable inbound SSH connections from global configuration mode.

**Note**

Disabling or enabling the SSH server does not impact existing sessions, but disabling it will prevent new SSH sessions from being established.

Feature Support for SSH

SSHv2 is the supported version of SSH, but not all features typically available with SSHv2 are supported on the Extreme devices.

The following encryption algorithms are supported:

- aes192-cbc
- aes192-ctr
- aes128-gcm-openssh
- aes128-ctr
- aes128-cbc
- aes256-cbc
- aes256-ctr: AES in Counter Mode (CTR) mode with 256-bit key
- aes256-gcm-openssh
- chacha20-poly1305-openssh

The following Hash-based Message Authentication Code (HMAC) message authentication algorithms are supported:

- hmac-sha1
- hmac-sha1-96
- hmac-sha1-etm-openssh
- hmac-sha2-256
- hmac-sha2-256-etm-openssh
- hmac-sha2-512
- hmac-sha2-512-etm-openssh

The following host keys are supported:

- ssh-rsa
- ECDSA

The following key exchange algorithms are supported:

- curve25519-sha256
- curve25519-sha256-libssh
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group-exchange-sha256
- diffie-hellman-group14-sha1
- diffie-hellman-group16-sha512
- diffie-hellman-group18-sha512
- diffie-hellman-group14-sha256
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521

SSH user authentication is performed with passwords stored on the device or on an external authentication, authorization, and accounting (AAA) server.

SSH Server Configuration

Extreme ONE OS supports Secure Shell (SSH) for secure remote management. The SSH server can be configured using the following command:

```
ssh-server NAME
```

Parameter	Type	Range	Default	Description
NAME	String	1-64	DEFAULT	SSH server instance name

Command Mode

```
config-system
```

Default Behavior

By default, an SSH server instance named DEFAULT is created and is automatically associated with 'mgmt-vrf'.

Configuration Commands

To configure an SSH server instance:

```
device(config-system) # ssh-server NAME
```

To configure the default instance:

```
device(config-system) # ssh-server  
device(config-system-ssh-server-DEFAULT) #
```

Verification Commands

To check the SSH server configuration:

```
device(config-system)# do show running-config system ssh-server
system
  ssh-server DEFAULT
    vrf mgmt-vrf
    enable
  ssh-server 1
  ssh-server xyz
```

To check the current running state:

```
device(config-system)# do show running-state system ssh-server
system
  ssh-server DEFAULT
    vrf mgmt-vrf
    enable
  ssh-server 1
  ssh-server xyz
```

SSH Host Key Preservation

To ensure continuity and security, SSH host keys are preserved during:

1. Firmware Upgrades: Keys are automatically backed up before the upgrade and restored afterward.
2. Full-Install Firmware Updates: The /etc/ssh folder is backed up and restored by default during the full installation process.

Optional: Disabling Key Preservation

You can choose to remove existing keys and disable preservation using the `no-preserve` parameter:

```
system firmware fullinstall <url path of firmware
file> [no-preserve]
```



Note

Using the `no-preserve` option will remove existing SSH host key.

Connecting to an Extreme Device with SSH

An SSH session allows you to access a device remotely using port 22.

1. Establish an SSH session connection to the Extreme device.

```
device# ssh admin@192.0.2.0
```

The example establishes an SSH session to the device with the IP address of 192.x.x.x.

2. Enter yes if prompted.

```
The authenticity of host '192.0.2.0 (192.0.2.0)' can't be established.
RSA key fingerprint is 9f:83:62:cd:55:6c:b9:e8:1d:79:ab:b4:04:f4:f6:2a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.0.2.0' (RSA) to the list of known hosts.
admin@192.0.2.0's password:
```

```
SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.
```

```
Welcome to the Extreme ONE OS Software
```

```
admin connected from 192.0.2.0 using ssh on device
device#
```

**Note**

The default admin login name is admin.

As a best practice, change the default account password when you log in for the first time.

Enable or Disable SSH Service

Follow this procedure to enable or disable SSH service.

1. In EXEC mode, enter global configuration mode.

```
device# configure terminal
```

2. Disable SSH service on the device.

```
device(config)# system
device(config-system)#
device(config-system)# ssh-server
device(config-system-ssh-server-DEFAULT)# no enable
device(config-system-ssh-server-DEFAULT)#
```

Disabling the SSH service will block new sessions, but existing sessions will remain unaffected.

3. Enable SSH service on the device.

```
device(config-system-ssh-server-DEFAULT)# enable
```

4. (Optional) The SSH Server can be restarted on all VRF instances using **ssh-server restart** command.

Configuring Terminal Session Parameters

To set the parameters, perform the following steps:

Set the timeout length.

```
device# terminal timeout 3600
```

This example sets the timeout of 3600 seconds (60 minutes) for the terminal session.

**Note**

Specifying a value of 0 allows the terminal session to stay open until the device is rebooted or the connection is terminated by other means.

Configuring Login Banner

The Extreme device can be configured to display a greeting message on user terminals as a banner when they enter the Privileged EXEC CLI level or access the device through Telnet/SSH. The Login banner displays a message before the user logs into the device.

Complete the following steps to set and display a login banner.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
device(config)# system
device(config-system)#
```

2. Configure the login banner.

```
device(config-system)# login-banner welcome
```

The banner can be up to 2048 characters long.

You can use the **no login-banner** command to remove the banner.

3. Verify the configured banner.

```
device(config-system)# "do show running-config system banner
system
login-banner welcome
!"

device(config-system)# do show running-state system banner
system
login-banner welcome
!"
device(config-system)#
```

The configured banner is displayed.

```
login-banner Welcome !
device(config-system)#
```

The following example is the configuration of the previous steps.

```
device# configure terminal
device(config)# system
device(config-system)# login-banner Welcome
```

Configuring MOTD Banner

The Extreme device can be configured to display a greeting message on user terminals as a banner when they enter the Privileged EXEC CLI level or access the device through Telnet/SSH. The MOTD-banner (Message of the Day) displays a message after the user logs into the device.

Complete the following steps to set and display a MOTD banner.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
device(config)# system
device(config-system)#
```

2. Configure the MOTD banner.

```
device(config-system)# motd-banner welcome!
```

The banner can be up to 2048 characters long. You can use the **no motd-banner** command to remove the banner.

3. Verify the configured banner.

```
device(config-system)# do show running-config system banner
system
```

```

login-banner hellow
motd-banner welcome!

device(config-system)# do show running-state system banner
system
login-banner hellow
motd-banner welcome!
device(config-system)#

```

System Logging Configuration

For comprehensive information on command syntax and parameters on logging configuration, see *Extreme ONE OS Switching v22.2.0.0 Command Reference Guide*.

The following table describes a list of commands used to configure system logging:

Command	Description
Service Severity Level: service [NAME] severity <emergency alert critical error warning notice info debug>	Configure Syslog logging severity level for a service (default is info).
Console/Monitor Severity Level: [no] <console monitor> severity <emergency alert critical error warning notice info debug>	Enable/disable displaying Syslog messages on console or SSH terminal (default is disabled, with default severity set to error).
Remote Server Configuration: [no] remote-server NAME	Configure remote server to forward Syslog messages, with options for transport mechanisms, remote port, secure forwarding, source address, and VRF.
Display Logging Information: <ul style="list-style-type: none"> show logging raslog show system logging facility show system logging remote-server show logging audit config firmware security 	<ul style="list-style-type: none"> Display all Syslog messages in time order. Display the severity level configured for each facility (service). Display remote server configuration details. Display audit logs for config, firmware, or security-related activities.
Monitoring: monitor <start stop>	Start or stop displaying Syslog messages on SSH-connected terminal (if enabled).
show running-config system logging For example, <pre> device# show running-config system logging console On Console session facility Facility information monitor On Monitor session remote-server Remote-server information device# </pre>	Display the current logging configuration, including: <ul style="list-style-type: none"> Console session facility information Monitor session information Remote server information

Ethernet Management Interfaces

The management Ethernet network interface provides management access, including direct access to the device CLI. You must configure at least one IP address using a serial connection to the CLI before you can manage the system. You can either configure static IP addresses, or you can use a Dynamic Host Configuration Protocol (DHCP) client to acquire IP addresses automatically. For IPv6 addresses, both static IPv6 and stateless IPv6 autoconfiguration are supported.



Important

Setting static IPv4 addresses and using DHCP are mutually exclusive. If DHCP is enabled, remove the DHCP client before you configure a static IPv4 address. However, this does not apply to IPv6 addresses.

Configuring Static Management Port

Before you configure a static address, connect to the device through the serial console. To configure static management port, perform the following steps:

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
device(config)#
```

2. Access the interface management mode for the management interface.

```
device(config)# interface management 0
device(config-if-mgmt-0)#
```

This interface uses the default mgmt-vrf VRF.

3. Configure the IP address for the management interface.

```
device(config-if-mgmt-0)# ipv4 address

    A.B.C.D/M   IPv4 address
dhcp           DHCP

device# conf
device(config)# interface management 0
device(config-if-mgmt-0)# ipv4 address 192.x.x.x/xx
```

Use the ? symbol to display help for a command

4. Configure the static route.

```
device(config)# vrf mgmt-vrf
device(config-vrf-mgmt-vrf)# static-route 0.0.x.x/0 192.x.x.0
```



Note

If you are going to use an IPv6 address, configure the address.

```
device# conf
device(config)# int ma 0
device(config-if-mgmt-0)# ipv6 address 2001::1/64
```

5. Verify the configuration.

```
device# show running-config interface management 0
interface management 0
    ipv4 address 1xx.0.x.0/xx
    no shutdown
```

```
!
device#
```

Displaying the Management Interface

You can display the information about the management interface on the device. If an IP address has not been assigned to the network interface, you must connect to the CLI using a console session on the serial port. Otherwise, connect to the device through Telnet or SSH.

```
device# show interface management 0

management 0 Admin state UP      Operational state UP
MTU 1514 bytes
Hardware is Ethernet  mac address b4:a3:bd:0e:00:00
Current Speed  1G, Duplex: Full
Description: Management 0
DHCPv4 Enabled [ 10.38.60.139/24 ]
DHCPv6 Disabled

Redundancy Mode: (Dual OOB)
  active-path: extMgmt 1
  primary-path: extMgmt 1
  all-paths: extMgmt 1, extMgmt 2

Statistics
  LastClear: 0s
Input:
  Total pkts: 1038
  Discard pkts: 414
  Errors pkts: 0
  CRC Errors: 0
  MCast pkts: 992
  Octets: 52080
device#
```

show interface extMgmt 1/2 (OOB port)

```
device# show interface extMgmt 1

extMgmt 1 Admin state UP, Oper state UP
Interface index is 1090519072 (0x41000020)
MTU 9216 bytes
Hardware is Ethernet  mac address b4:a3:bd:0e:00:19
Current Speed  1G
Description: External Management 1

device# show interface brief
Flags:  M - Redundant Management  P - Performance-Path
Number of interfaces 37
Port      Mtu      Admin-State Oper-State Speed    Ifindex    Description
-----
ExtMgmt 1 9216     DOWN       DOWN     1G       0x41000020  ExtMgmt 1
ExtMgmt 2 9216     DOWN       DOWN     10G      0x41000040  ExtMgmt 2
Int 0     9216     UP         DOWN     20G      0x21000000  Internal 0
Int 1     9216     UP         DOWN     20G      0x21000020  Internal 1
Eth 0/1   9216     DOWN       DOWN     400G     0x1000020   Ethernet 0/1
Eth 0/2   9216     DOWN       DOWN     400G     0x1000040   Ethernet 0/2
device#
```

show sysinfo media detected

```
device# show sysinfo media detected
```

----- Management Media						
S/C	Qual	Optical	Type	PartNum	SerialNum	
Vendor		Description				
extMgmt 1	Yes	No	SFP	FCLF8522P2BTL-EX	NBHC437	FINISAR
CORP.	1G BASE-T	SFP 100m				
extMgmt 2	No	Yes	SFP_PLUS	57-0000075-01	AAF210440000GHS	
BROCADE		10GE SR SFP+				

```
device#
```

show sysinfo media interface <extMgmt 1>

```
device# show sysinfo media interface extMgmt 1
```

Media Information:

```

    Interface: extMgmt 1
      Cage: 33
      Slot: 0
      Qualified: Yes
      Optical: No
      State: Inserted
      Module Type: SFP
      Part Number: FCLF8522P2BTL-EX
      Serial Number: NBHC437
      Vendor: FINISAR CORP.
  Supports Breakout: No
    Description: 1G BASE-T SFP 100m
      Channels: 0
      Datecode: 2024-05-09
      BaudRate: 12 (units 100 megabaud)
      LengthSmfKm: 0 (units km)
      LengthSmfM: 0 (units 100 meters)
      Length50UmM: 0 (units 10 meters)
      Length625UmM: 0 (units 10 meters)
      Wavelength: 0 (units nm)
      LengthCopper: 100 (units 1 meter)
      VendorRev: A
      Options: 0012
      BrMax: 0
      BrMin: 0
device#
```

show running-config interface <extMgmt 1>

```
device# show running-config interface extMgmt 1
```

```

interface extMgmt 1
  no shutdown
!
device#
```


Configuring an IPv6 Address on the ONE OS Platform

Following are the basic pre-requisites for configuring an IPv6 address on the ONE OS platform:

- PC connected to the serial port of the device.
- IPv6 network assignment with a netmask and router address from the network administrators. This will generally be a /64 network.



Note

If you are provided an IPv6 prefix with a /65 to /128 netmask, assign the addresses according to your network administrator's direction, and do NOT follow this procedure.

Follow the procedure to configure IPv6 addresses.

1. To configure the IPv6 address, run the following command:

```
device(config)# interface management 0
device(config-if-mgmt-0)# ipv6 address 2001::100/64
device(config-if-mgmt-0)# ipv6 address 2001::101/64
device(config-if-mgmt-0)# ipv6 address 2002::100/64
device(config-if-mgmt-0)# ipv6 address 2002::101/64 secondary
device(config-if-mgmt-0)# ipv6 address 2003::/64 eui-64
device(config-if-mgmt-0)# ipv6 address fe80::/64 eui-64
device(config-if-mgmt-0)# interface ethernet 0/2
device(config-if-mgmt-0)# ipv6 address fe80::f264:26ff:fe5:c805/64
```

2. To enable IPv6 address DHCP configuration, run the following command:

The DHCP server will provide only the IPv6 address, but not the Gateway Address.

```
device(config-if-eth-0/1)# interface ethernet 0/1
device(config)# ipv4 address dhcp
device(config)# ipv6 address dhcp
```

3. To enable the auto-configuration of IPv6 address, run the following command:

```
device(config)# interface ethernet 0/1
device(config-if-eth-0/1)# ipv6 address auto-config
```

Port Management

The 8730-32D port management offers:

- Total support for 32 ports at 400G
- Flexible breakout configurations of 4x100G, 4x10G, 4x25G on all 32 ports
- Forward Error Correction (FEC) enabled for 25G, 100G, and 400G speeds

Extreme ONE OS 100G Ports

For fixed form factor Extreme ONE OS devices, all 100Gb/40Gb interfaces default to 100Gb mode.

You can configure 40G mode using the speed 40000 command from the interface configuration mode. Each 100G port also supports 4x25G and 4x10G breakout configurations.

Link Fault Signaling

The ONE OS platform supports Link Fault Signaling (LFS) detection for interface types of 40G, 100G, and 40G breakout ports. It detects local and remote faults.

Link fault signaling monitors the status and health of the link. You can enable or disable this feature in the front physical interface settings. By default, link fault signaling is enabled on all front physical interfaces.

Show Interface Ethernet

Use the **show interface ethernet <interface>** command to view the link fault signaling status, including, admin and operational state, interface index, MTU, hardware and MAC address, and speed and FEC mode

```
device# show interface ethernet 0/1
ethernet 0/1 Admin state UP ,Oper state DOWN
Interface index is 16777248 (0x1000020)
MTU 9216 bytes
Hardware is Ethernet mac address 20:9e:f7:7e:20:04
Current Speed 100G
FEC mode disabled
Link Fault Signaling: ON
Link Fault Status: Local Fault
Auto Negotiation: OFF
```

The link fault status will be one of the following:

- No Fault: Port is administratively down
- No Fault: Port is operationally up, and no faults are set
- Local Fault: Port does not detect laser
- Remote Fault: Port receives remote fault

Show Counter Link-Fault Signaling

Use the **show counters link-fault-signaling** command to view the number of local and remote faults on the port along with the last fault seen time.

```
device# show counters link-fault-signaling
```

Port	Local-Fault-Count	Last-Local-Fault	Remote-Fault-Count	Last-Remote-Fault
=====	=====	=====	=====	=====
Eth 0/1	3	2023-05-24T06:54:48Z	0	NA
Eth 0/2	0	NA	0	NA
Eth 0/3	0	NA	0	NA
Eth 0/4	1	2023-05-24T04:56:57Z	0	NA
Eth 0/5	0	NA	0	NA
Eth 0/6	0	NA	0	NA
Eth 0/7	0	NA	0	NA
Eth 0/8	0	NA	0	NA

Show Link-Fault Signaling

Use the **show link-fault-signaling** command to view the link fault signaling status for each front port.

```
device# show link-fault-signaling

Rx and Tx link-fault-signaling status

Port          Link-Faults
=====
Eth 0/1       ON
Eth 0/2       OFF
Eth 0/3       ON
```

Dual OOB Redundant Management Port Configuration

The switch has two SFP+ management ports (Primary and Backup) connected to the Extreme 8730 device. These ports support Active-Standby mode with link failover.

CLI Commands

- **show interface management 0**
- **show sysinfo media detect**
- **show sysinfo media interface <extMgmt 1>**
- **show interface <extMgmt 1>**
- **show running-config interface <extMgmt 1>**
- **<td4-config> interface <extMgmt 1>**

For more details on command syntax and parameters, see *Extreme ONE OS Switching Command Reference Guide*.

Limitations

Known limitations include:

1. If the primary link is plugged in while the secondary link is active, and then the secondary link is removed within 5 seconds, the active mode may not switch to the primary link.
2. Interface counter show/clear commands not supported on the external management port "extMgmt <1/2>".
3. IP configuration not supported on external management ports as these ports are L2 interfaces.

4. MTU configuration not supported on external management ports
5. When Management 0 interface is up or down, it has a fixed speed of 10G. The speeds of Mgmt 1 and Mgmt 2 are reflected on their respective interfaces: extMgmt 1 and extMgmt 2.

**Note**

In a specific scenario where the IAH TPVM starts with both external management (extMgmt) links in a down state, a software limitation causes an issue. The system implements a sync logic that inherits the link status of extMgmt on the TAP IAH VM interface, which is created as a slave to ma1. This sync logic results in a flap on ma1, causing it to go up and then down. This behavior is a current software limitation.

Interface Ethernet Ports

All Extreme device ports are pre-configured with default values that allow the device to be fully operational at initial startup without any additional configuration. In some configuration scenarios, changes to the port parameters may be necessary to adjust to attached devices or other network requirements.

Displaying Device Interfaces

The device supports Ethernet, loopback, management, and virtual Ethernet interfaces (VEs).

Enter the **show running-config interface** command to display the interfaces and their status.

The following example displays the Ethernet interfaces on the device and are identified by the port number.

For example, the notation 0/8 indicates port 8 on a device.

```
device# show running-config interface ethernet
interface Ethernet 0/1
  no shutdown
!
interface Ethernet 0/2
  channel-group 101 mode active type standard
  lacp timeout long
  no shutdown
!
interface Ethernet 0/3
  channel-group 101 mode active
<cr>

interface Ethernet 0/4
  shutdown
!
interface Ethernet 0/5
  shutdown
!
interface Ethernet 0/6
  shutdown
!
```

```
interface Ethernet 0/8
  channel-group 143 mode active
<cr>
!
interface Ethernet 0/9
  shutdown
!
```

System Clock



Note

You can set the system clock if there are no NTP servers configured. Otherwise, an active NTP server, if configured, automatically updates and overrides the system clock.

Setting the Clock

Setting the clock is not required when NTP is configured and the clock is synchronized to an external NTP server.

To set the clock, perform the following step:

1. In EXEC mode, run the following command:

```
device# clock set 2025-07-18T23:00:00
device# show clock
2025-07-18 23:00:04 UTC +0000
device#
```

2. Change the time zone in global configuration mode.

```
device# configure terminal
device(config)# system
device(config-system)# clock timezone Australia/Melbourne
device(config-system)# end
```

3. Verify the time zone.

```
device# show running-config system clock
system
  clock timezone  Australia/Melbourne
!
```

This example changes the time zone to the region of Australia and the city of Melbourne.

Management VRFs

All management services on the Extreme device are VRF aware. The management services can select a particular VRF to reach a remote server based on a VRF. The VRFs are management (mgmt-vrf), default (default-vrf), and user-defined VRFs.

By default, the device creates a VRF for management named mgmt-vrf and, all manageability services are accessible through this VRF. Multiple instances of IP services can be instantiated in multiple VRFs. For example, SSH can be in more than one VRF.

VRF Reachability

VRF reachability indicates the details of the VRF for servicing requests from the clients. It also indicates the clients specifying the VRF for reaching a source to ensure that the management packets are serviced or routed in a server VRF domain.

These two types of reachability services are also referred to as device-initiated and server-based services.

VRF Reachability for Device-Initiated Services

The following table lists the device-initiated services and associated commands that VRF reachability supports.

Service	VRF-related command	Additional information
LDAP	vrf NAME <String: 1-64 character> vrf name	Default vrf is mgmt-vrf . User-defined VRF is optional.
Logging server	vrf NAME <String: 1-64 character> vrf name	Uses TCP UDP IPv4 or IPv6.
NTP	ntp server <server ip address> Configure VRF under NTP server: vrf VRFNAME <STRING 1-64> Name of the VRF instance	Uses NTP UDP IPv4 and IPv6.
RADIUS	server-group [tacacs+ radius ldap] Configure user defined vrf: vrf NAME <String: 1-64 character> vrf name	Uses UDP and TCP IPv4 or IPv6
Copy file	device# copy file URL Source URL disk://config-file/<file-name> disk://coredumps/<file-name> disk://firmware/<file-name> disk://iah/<file-name> disk://tech-support/<file-name> usb://<file-name> scp:// <username>[:<password>]@<host>: [port]/<filepath>	Uses TCP IPv4 or IPv6. By default, mgmt-vrf is used and a user-defined VRF is optional.
TACACS+	server-group tacacs+-server-address Configure user defined vrf: vrf NAME <String: 1-64 character> vrf name	Uses TCP IPv4 or IPv6

All these implementations use forward referencing of the VRF name in the **vrf defined** option, unless noted. At runtime when making the socket connection, the VRF ID by name must be resolved. If it does not resolve, it will result in a connection error.

VRF Reachability for Server-Based Services

The server services running on the Extreme device must listen to the requests in all the VRFs or a specified VRF and send the response back to the client in the same VRF where the request arrived. Thus, the services can come through any in-band interface bound to any VRF.

Each server-based service can have a maximum of 32 VRF instances; one mgmt-vrf, one default-vrf, and 30 user-defined VRFs. The following table lists the server services and associated commands that VRF reachability supports.



Note

You can configure the SNMP server to listen on any Virtual Routing and Forwarding (VRF) instance, but it is disabled by default.

Table 4: Server-based services and associated commands that VRF reachability supports

Service	VRF-related command	Additional information
gRPC	grpc-server NAME device(config-system-grpc-server-test)# vrf VRFNAME <String: 1-64 character> vrf name	By default, the gRPC server is not configured. You need to configure and enable it. This command is used to create a gRPC server instance.
SSH	ssh-server NAME #config-system-ssh-server- <ssh_name>	By default, ssh-server 'DEFAULT' instance is created and is associated with 'mgmt-vrf'. A user-defined VRF is optional.
Telnet	telnet server NAME config-system-telnet-server- <server_name>#	By default, the Telnet service is not configured, and you need to configure and enable it. This command is used to create a telnet server instance. If NAME is not specified, 'DEFAULT' will be used as the name of the telnet-server instance.
SNMP notification	config-system-snmp-server-vrf-<vrf-name>-host-<host-name-community-or-user-name>	Uses SNMP UDP IPv4 or IPv6. By default, mgmt-vrf is used to send the SNMP notifications.

Telnet and SSH Limitations and Considerations

Telnet and SSH limitations and considerations are as follows:

- You cannot remove mgmt-vrf from the SSH and Telnet services.
- Telnet and SSH server can be enabled on a maximum number of 32 VRFs.
- SSH and Telnet Services started on VRF context is applicable for both IPv4 and IPv6 addresses.

- A maximum of 32 user logins are allowed on the device. These sessions are a cumulative count of login sessions through SSH and Telnet across all the configured VRFs.
- Inter-VRF route leaking is not supported for SSH and Telnet. When you try to use any of these services to access a leaked VRF (user-vrf), the connection is refused. In addition, the access service ceases to function correctly on the local VRF (mgmt-vrf) and you must restart it; for example, to restart the SSH service on the local VRF, run the **ssh server restart** command.

MAC Address Aging

MAC addresses that are dynamically learned are stored in MAC address table. The MAC address aging feature provides a mechanism to flush out the dynamic MAC addresses that remain inactive for a specified period.

The aging time of dynamic MAC address entries can be configured using the **mac-address-table aging-time** command. You can disable the MAC address aging by specifying the aging time as 0 (zero). The MAC aging time can be configured to a value from 60 through 38400 seconds. By default, the aging time of dynamic MAC address entries is 1800 seconds. The configured MAC aging time is applied to all MAC addresses in the system.



Note

The MAC address aging configuration per bridge-domain is not supported.

The following is an example configuration of MAC address aging (MAC forwarding table aging time):

```
device(config)# system
device(config-system)# global
device(config-system-global)# mac-address-table aging-time
(60-38400) Aging time in seconds (default = 1800)
0          Disable aging
device(config-system-global)# mac-address-table aging-time
```

BGP Protocol Event Monitoring and Notification

Border Gateway Protocol (BGP) is a crucial Internet routing protocol that enables traffic exchange between Autonomous Systems (AS) and ensures loop-free routing. An AS is a network collection sharing common routing and administrative characteristics. Within an AS, Interior Gateway Protocols (IGPs) are used, while Exterior Gateway Protocols (EGPs) connect different AS. Only the Extreme 8730 hardware platforms are supported.

Supported Functionalities

- gNMI notifications
- RAS trace logs
- BGP SNMP trap for Enterprise MIB and Standard MIB

BGP Enterprise and Standard MIB Notifications

BGP reports significant events to the message bus when a BGP session changes state to Established or experiences backward transitions. These BGP traps contain session information within their payload/Varbind. The BGP Enterprise and Standard MIB define specific trap OID and Varbind lists for this purpose.

BGP Standard-MIB Notifications are sent for the peers of IPv4 types.

Table 5: BGP Standard-MIB Notifications

Trap Name and OID	Varbinds	Description
bgpEstablishedNotification 1.3.6.1.2.1.15.0.1	bgpPeerRemoteAddr bgpPeerLastError bgpPeerState	The bgpEstablishedNotification event is generated when the BGP FSM enters the established state.
bgpBackwardTransNotification 1.3.6.1.2.1.15.0.2	bgpPeerRemoteAddr, bgpPeerLastError, bgpPeerState	The bgpBackwardTransNotification event is generated when the BGP FSM moves from a higher numbered state to a lower numbered state.

The BGP Enterprise-MIB Notifications are sent for the peers of IPv6 types.

Table 6: BGP Enterprise-MIB Notifications

Trap Name and OID	Varbinds	Description
extremeBGP4V2EstablishedNotification 1.3.6.1.4.1.1916.1.51.0.1	extremeBgp4V2PeerState extremeBgp4V2PeerLocalPort extremeBgp4V2PeerRemotePort extremeBgp4V2PeerRemoteAddr	The extremeBGP4V2EstablishedNotification event is generated when the BGP FSM enters the established state.
extremeBGP4V2BackwardTransitionNotification 1.3.6.1.4.1.1916.1.51.0.2	extremeBgp4V2PeerState extremeBgp4V2PeerLocalPort extremeBgp4V2PeerRemotePort extremeBgp4V2PeerLastErrorCodeReceived, extremeBgp4V2PeerLastErrorSubCodeReceived, extremeBgp4V2PeerLastErrorReceivedText extremeBgp4V2PeerRemoteAddr	The extremeBGP4V2BackwardTransitionNotification event is generated when the BGP FSM moves from a higher numbered state to a lower numbered state.

gNMI Notifications

BGP neighbor config or state changes are published to the gNMI path `network-instances/network-instance[name=]/protocols/protocol[identifier=BGP][name=bgp]/bgp/neighbors/neighbor[neighbor-address=]/`

```
+--rw network-instances
  +--rw network-instance* [name]
    . . .
    +--rw protocols
      | +--rw protocol* [identifier name]
      | +--rw bgp
      . . .
        | +--rw neighbors
        | | +--rw neighbor* [neighbor-address]
        | | | +{}rw neighbor-address{-} > ../config/neighbor-address
        . . .
        | | +--ro state oc-inet:as-number
        | | | +--ro session-state?
        enumeration
        | | | +--ro last-established? oc-
        types:timeticks64
        | | | +--ro established-transitions? oc-
        yang:counter64
        | | | +--ro supported-capabilities*
        identityref
        | | | +--ro messages
        | | | | +--ro sent
        | | | | | +--ro UPDATE? uint64
        | | | | | +--ro NOTIFICATION? uint64
        | | | | | +--ro last-notification-time? oc-
        types:timeticks64
        | | | | | +--ro last-notification-error-code? identityref
        | | | | | +--ro last-notification-error-subcode? identityref
        | | | | +--ro received
        | | | | | +--ro UPDATE? uint64
        | | | | | +--ro NOTIFICATION? uint64
        | | | | | +--ro last-notification-time? oc-
        types:timeticks64
        | | | | | +--ro last-notification-error-code? identityref
        | | | | | +--ro last-notification-error-subcode? identityref
```

RAS Logs

The following are Session UP/Down Raslogs:

```
2025-01-16 11:04:36.7728 bgp[15]: {"Level":"info","LogID":9008,"Topic":2,"VRF":"default-vrf","Neighbor":"192.x.x.x","Reason":"ADMIN-DOWN","Msg":"Session DOWN"}
2025-01-16 11:04:36.7729 bgp[15]: {"Level":"info","LogID":9008,"Topic":2,"VRF":"default-vrf","Neighbor":"10.x.x.x","Reason":"ADMIN-DOWN","Msg":"Session DOWN"}
2025-01-16 11:06:29.1857 bgp[15]: {"Level":"info","LogID":9008,"Topic":2,"VRF":"default-vrf","Neighbor":"10.x.x.x","Msg":"Session UP"}
2025-01-16 11:06:31.8469 bgp[15]: {"Level":"info","LogID":9008,"Topic":2,"VRF":"default-vrf","Neighbor":"10.x.x.x","Msg":"Session UP"}
```

CLI Commands

Use this topic to learn about the BGP Clear, BGP Config, and BGP Show commands.

**Note**

For more information about commands and supported parameters, see Extreme ONE OS Switching Command Reference Guide.

1. Clear Commands

- clear bgp neighbor
- clear bgp vrf default-vrf routes l2vpn-evpn nvo <>
- clear bgp vrf default-vrf routes l2vpn-evpn nvo base route-distinguisher <ASNUMBER:ADMINNUMBER> or <IPV4-ADDRESS:ADMINNUMBER>
- clear bgp routes l2vpn-evpn route-type ARP
- clear bgp routes l2vpn-evpn route-type IMR
- clear bgp routes l2vpn-evpn route-type MAC
- clear bgp routes l2vpn-evpn route-type ND
- clear bgp routes l2vpn-evpn route-type prefix ipv4
- clear bgp routes l2vpn-evpn route-type prefix ipv6

2. Configuration Commands

- router bgp
 - ipv4-unicast/ipv6-unicast address-families
 - activate
 - add-paths
 - graceful-restart
 - network
 - next-hop-enable-default
 - next-hop-recursion
 - prefix-independent-convergence
 - send-default-route
 - use-multiple-paths
 - l2vpn-evpn address-family:
 - activate
 - graceful-restart
 - retain-route-target-all
 - use-multiple-paths
- as-notation confederation
- confederation member-as
- graceful-restart [global]
 - helper-only [global]
 - restart-time [global]
 - stale-route-time [global]

- Instance type
 - address-family (VRF)
 - L3VNI (VRF)
 - local-as
 - peer group
 - address-family
 - allow-own-as
 - auth-password
 - cluster-id
 - description
 - ebgp-multihop
 - enable-bfd
 - fast-external-failover
 - graceful-restart
 - listen-range >> this cli can be cfd as "listen-range" / "listen-range <> listen-limit <>"
 - local-as-forced
 - neighbor
 - nvo
 - remote-as
 - route-reflector-client
 - shutdown
 - timers
 - update-source
 - router-id
 - use-multiple-paths
3. Show Commands
- show bgp vrf <vrf_name> l2vpn-evpn instance/tunnels
 - show bgp vrf <vrf_name> neighbor
 - show bgp vrf <vrf_name> routes
 - show bgp vrf <vrf_name> summary

Support for OpenConfig Telemetry

OpenConfig is a vendor-neutral, model-driven network management specification, where data models are used for both configuration as well as retrieving operational state of the network across platforms.

OpenConfig proposes to use gNMI (gRPC Network Management Interface) framework as the network management protocol for configuration, data retrieval, and real-time network monitoring support. It provides mechanisms to modify and retrieve configuration information from target devices. It also provides the ability to generate and control telemetry streams from these target devices to a data collection system.

Extreme ONE OS enables the retrieval of operational data and configuration for all supported YANG modules and paths on the device.

For each module, of the large amount of information that can be fetched, we support a small set. The following section lists the operational state information that can be fetched for each module.



Note

Use the following command to configure the certificate expiry alert. This applies to app certificates and only those that are associated with a protocol (for example, grpc-server) .

```
device(config-system-cert-mgr)# expiry-alert ?
  info      Info severity log expiry alert period
  minor     Minor severity log expiry alert period
  major     Major severity log expiry alert period
  critical   Critical severity log expiry alert period

device(config-system-cert-mgr-exp)# info?
DAYS <1-90> Days before certificate expiry logging

device(config-system-cert-mgr-exp)# minor?
DAYS <1-90> Days before certificate expiry logging

device(config-system-cert-mgr-exp)# major?
DAYS <1-90> Days before certificate expiry logging

device(cconfig-system-cert-mgr-exp)# critical?
DAYS <1-90> Days before certificate expiry logging
```

OpenConfig-BGP Yang Module

For the *BGP Module*, the following operational state information can be fetched:

```
module: openconfig-bgp
path: /bgp/neighbors/neighbor[neighbor-address=<nAddress>]

module: openconfig-bgp
+--rw bgp
| +--rw neighbors
| | +--rw neighbor* [neighbor-address]
| | | +--rw neighbor-address -> ../config/neighbor-address
| | | | +--ro neighbor-address? oc-inet:ip-address
| | | +--rw transport
| | | | +--ro state
| | | | +--ro local-port? oc-inet:port-number
| | | | +--ro remote-port? oc-inet:port-number
| | | +--rw use-multiple-paths
| | | | +--ro state
| | | | | +--ro enabled? boolean
| | | +--rw ebgp
| | | | +--ro state
| | | | +--ro allow-multiple-as? boolean
```

OpenConfig-Platform Yang Module

For the *Platform Module*, the following operational state information can be fetched:

```
module: openconfig-platform
path: /components/component[name=<cmpntName>]/state

+--rw components
  +--rw component* [name]
    +--rw name                               -> ../config/name
    +--ro state
      | +--ro name?                          string
      | +--ro id?                            string
      | +--ro mfg-name?                       string
      | +--ro hardware-version?              string
      | +--ro firmware-version?              string
      | +--ro software-version?              string
      | +--ro serial-no?                     string
      | +--ro part-no?                       string
      | +--ro removable?                     boolean
      | +--ro empty?                         boolean
      | +--ro parent?                        -> ../../config/name
      | +--ro temperature
      | | +--ro alarm-status?                 boolean
      | | +--ro alarm-threshold?              uint32
```

Securing OpenConfig Telemetry Connections

It is assumed that your infrastructure is set up with a gNMI client. Its configuration is beyond the scope of this document.

This topic describes the steps to secure incoming connections from gNMI clients. By default, the gNMI server on ONE OS listens on the insecure port 9339. To secure incoming connections, you must configure a port (range 1024-49151) on which the gNMI server listens on for incoming connections. The existence of this port configuration determines whether the gNMI server is listening for incoming connections in the secure or insecure mode.

The gNMI server runs on a gRPC server, which can be configured on the device using the CLI. The default port for the gRPC server is 443.

Zero Touch Provisioning (ZTP) over DHCP

Learn how to configure and implement ZTP over DHCP for Extreme ONE OS platforms. ZTP streamlines device setup by automatically downloading firmware and configuring devices, eliminating manual console login and configuration.

When a customer connects a new Extreme device to their network, ZTP automatically loads the correct firmware and configurations without manual intervention. To enable ZTP, the DHCP server must be configured with two specific DHCP options

ZTP Configuration

Configuring DHCP Option 66: HTTPS Server IP Address

To enable Zero Touch Provisioning (ZTP), DHCP Option 66 must be set up with the IP address of the HTTPS server.

This HTTPS server will store essential files, including:

- ZTP configuration files
- Device configuration files for configuration replay
- Firmware updates
- Python scripts with startup configurations

If the HTTPS connection fails, the device will attempt to download files using HTTP.

Configuring DHCP Option 67: ZTP Configuration File Path

To complete the Zero Touch Provisioning (ZTP) setup, DHCP Option 67 must be configured with the file path of the ZTP configuration file, which is stored on the specified HTTPS server.

Configuring DHCP Option 43: CA Certificate File for HTTPS Server

Optionally, DHCP Option 43 can be configured with a hex-encoded value containing the URL where the CA certificate can be retrieved, enabling secure communication with the HTTPS server specified in Option 66.

HTTPS Server Requirements

The HTTPS server hosts essential files for Zero Touch Provisioning (ZTP), including:

- ZTP configuration files
- Firmware updates
- Device configuration files
- Python scripts

If HTTPS connectivity fails, devices will attempt to download files using HTTP.

ZTP Configuration File

The ZTP configuration file is a crucial component that directs the Zero Touch Provisioning process. It contains:

- Firmware upgrade locations
- Device configuration file replay locations
- Python script locations

This ZTP configuration file supports three levels of configuration settings:

1. Global (Common): Applies to all devices
2. Group-Specific: Applies to a group of devices
3. Device-Specific: Applies to individual devices, identified by DHCP Client Identifier prefix or full identifier

The configuration file is parsed in the following order of priority:

1. Device-Specific
2. Group-Specific
3. Global (Common)

If an attribute is defined in multiple sections, the most specific section's value takes precedence.

**Note**

- Lines starting with "#" are treated as comments and ignored.
- Lines with empty values are considered valid.

Firmware Image

The latest device firmware will be hosted on the HTTPS server for installation during the ZTP process.

Device Configuration File

Customers can replay a set of startup configurations on their device. The replay process will continue regardless of success or failure.

Error logs for the replay process can be found at the `/var/data/ztp/log/` location.

This log file provides a record of any errors that occurred during the configuration replay.

Python Scripts

Python scripts required for automation or custom configuration tasks during the Zero Touch Provisioning (ZTP) process will be hosted on the same HTTPS server, enabling seamless execution and integration.

The response of the python script execution will be available at the `/var/data/ztp/log/` location.

ZTP Invocation

ZTP will get automatically triggered in the following scenarios:

1. Netinstall: After a netinstall, the switch reverts to its factory default state. Upon reboot, ZTP initiates automatically. This enables seamless deployment of devices shipped from the manufacturing site. Simply connect the devices to a preconfigured ZTP network, and the ZTP process will automatically upgrade firmware and configure settings.
2. Factory Reset: Running the "factory-reset" command restores the device to its factory default state, triggering an automatic reboot. Once the device is back online, ZTP starts automatically.

Without this feature, manual configuration would be required, involving:

- Connecting a console to the switch

- Configuring the switch management interface

This automated ZTP process simplifies large-scale switch deployments, saving time and effort.

ZTP DHCP Discovery Process

Upon arrival at the customer site, the device boots with its default configuration, enabling DHCP on the management interface. The device then initiates the following steps:

1. IP Address Request: The device sends a standard DHCP request to the DHCP server to obtain an IP address.
2. ZTP Configuration Request: Along with the IP address request, the device also requests:
 - DHCP Option 66: The IP address of the HTTPS server hosting the ZTP configuration files.
 - DHCP Option 67: The location of the ZTP configuration file, which guides the device path for Firmware, startup configuration file, python scriptReplay.
 - DHCP Option 43: A hex-encoded HTTP URL in TLV format, specifying the location of the HTTPS server's CA certificate.

These requests enable the device to discover and download the necessary configuration files for automated provisioning.

ZTP Client State Machine

The ZTP process unfolds as follows:

1. Device Boot: The device boots up in its factory default state.
2. ZTP DHCP Process: The device initiates the ZTP DHCP process, discovering the HTTPS server and boot file over the management interface using DHCP.
3. ZTP Configuration File Download: The device downloads the ztp.conf file from the HTTPS server specified in DHCP Option 66.
 - The device attempts to download the file using HTTPS. For HTTPS to succeed, the device must have the CA certificate of the HTTPS server which may be optionally available from DHCP server option 43.
 - If HTTPS fails, the device tries to download the ZTP configuration file using HTTP.
4. Firmware and Configuration Download: Based on the ztp.conf file, the device downloads:
 - Firmware
 - Startup configuration files
 - Python scripts
5. Device Configuration and Firmware Upgrade: The device
 - Upgrades or downgrades firmware
 - Executes the device startup configuration file
 - Executes Python scripts

6. ZTP Completion: Once the ZTP process is complete, it will not restart unless a factory reset is performed
7. ZTP Cancellation: If the ZTP process fails continuously, the user can cancel ZTP. However:
 - Once the ZTP process has started, cancellation is not allowed if the device is in the middle of a firmware upgrade or configuring the switch with remote server configurations.
 - To restart the ZTP process after cancellation, the user must execute the factory-reset command, which cleans up device configurations, reboots the device, and brings it back to ZTP mode.
8. ZTP Process Restart: If the device reboots or crashes during the ZTP process, the ZTP process will restart from step 3.
9. Configuration Lockdown: During the ZTP process, configuration changes are locked out until ZTP is completed or canceled. However,

Error Handling and Recovery

The Zero Touch Provisioning (ZTP) process includes robust error handling mechanisms:

- Network Issues or Firmware Unavailability: If firmware package downloads fail due to network problems or image unavailability, the ZTP process will automatically restart after a 1 min delay.
- Kernel Panic or Daemon Termination: If a kernel panic or daemon termination occurs during firmware download, the switch will reboot. Upon the next reboot, the ZTP process will retry, ensuring continuity and reliability.

ZTP CLI Commands

For complete information on ZTP CLI commands, such as ZTP Cancel, Factory Reset, show ztp dhcp status, and show ztp dhcp logs commands and syntax, see *Extreme ONE OS Switching Command Reference*.

Debugging Logs for Zero Touch Provisioning (ZTP)

To facilitate troubleshooting and debugging, ZTP logs are available at the following locations:

1. ZTP Base Logs: /tmp/tierra/trace/ztp-base/ztp-base-RASTrace.log
2. Python Script Logs: /var/data/ztp/log/pythonZtp.log
3. Switch Startup Configuration Logs: /var/data/ztp/log/configReplay.log
4. Consolidated ZTP Logs: /var/log/ztp/log/ztp.log

This log file is used by the **show ztp logs** command, providing a centralized view of ZTP-related logs.

Config and DRC (Drift & Reconcile) Tracking

The DB Audit feature enables the tracking of configuration changes on devices, facilitating various operational and reconciliation use cases. By providing a device-side

RPC, it offers a robust method for retrieving metadata on the most recent configuration change. This capability supports key services, including:

- Drift detection
- State synchronization
- Configuration tracking

By leveraging this feature, organizations can enhance their change management and monitoring capabilities

Key Features and Capabilities

This feature provides a robust tracking mechanism for configuration changes, enabling:

- **Reliable Change Tracking:** Accurately tracks and retrieves metadata for the latest configuration and state updates.
- **Device-Side RPC:** Implements a device-side RPC to fetch the latest version and timestamp for configuration and state changes.
- **Automatic Versioning and Timestamp Updates:** Automatically updates version and timestamp for any database changes made via CLI, gNMI, or other services.
- **Drift Detection and Reconciliation:** Enables external systems to detect configuration mismatches and trigger corrective actions.
- **Persistent Versioning:** Configuration version persists across reboots, while state version resets upon restart.

Persistence Handling in CDB vs SDB Reload

The system handles persistence across reboots and installations as follows.

Scenario	CDB Handling	SDB Handling
ONIE Install / System Full Install / Factory Reset	The version number resets to 0, but retains some values due to default sysconfig settings (x, a non-zero value).	Version resets to 0.
Restart After Installation	On restart, the version number becomes 0 + x, reflecting the retained sysconfig values.	Version resets to 0.
Reload / Reboot	After a reboot, the system retrieves the persisted version number from the config database into local memory.	Version resets to 0.

Scenario	CDB Handling	SDB Handling
Copy Default Config to Running Config	Before flushing the configuration, the system stores YANG paths and corresponding data in key-value pairs in a file. This file includes the version number.	Version resets to 0.
Restore Configuration on Reboot	On reboot, the system restores the version number from the stored file.	Version resets to 0.



ONE OS and Linux Shell Interoperability

[Overview](#) on page 53

[Accessing the Linux Shell from ONE OS](#) on page 53

[Escalating Linux Permissions](#) on page 54

[Saving and Appending Show Command Output to a File](#) on page 54

[Logs of Linux Shell Activities](#) on page 55

[Firmware Fullinstall Support](#) on page 55

Overview

As an Extreme ONE OS user with admin permissions, you can perform the following tasks:

- Running permitted Linux commands and scripts from the Extreme ONE OS CLI
- Accessing the Extreme ONE OS Linux shell, and:
 - Running permitted Linux commands and scripts.
 - Running Extreme ONE OS configuration and show commands.
 - Running scripts that contain multiple Extreme ONE OS commands.

Limitations

- Do not modify Extreme ONE OS user accounts from the Linux shell.

Accessing the Linux Shell from ONE OS

Inside the Extreme ONE OS Linux shell, you can execute commands that do not require root permissions.



Note

For details on using Python commands to launch an interactive Python3 shell, refer to the *Extreme ONE OS v22.2.0.0 Command Reference Guide*.

1. To access the Extreme ONE OS Linux shell, enter the **start-shell** command.

```
device# start-shell
[admin@device]#
```

2. Enter Linux commands and run scripts as needed. You can also run Extreme ONE OS commands from the Linux shell.

```
[admin@device]# cli_shell -c "show version"
```

3. To exit the shell and return to the Extreme ONE OS CLI, enter **exit**.

```
[admin@device]# exit
exit
device#
```

Upon exiting, the following message appears and you return to the Extreme ONE OS CLI prompt.

```
exit
device#
```

Escalating Linux Permissions



Caution

A user with Extreme ONE OS Linux-shell root permissions can—unintentionally or maliciously—execute commands that can render the ONE OS inoperable.

1. From the Extreme ONE OS CLI prompt, enter **start-shell** to access the Extreme ONE OS Linux shell.

```
device# start-shell
[admin@device]# sudo bash
[root@device]#
```

You can now execute commands that do not require root permissions.

2. Enter Linux commands and run scripts as needed.

You can also run Extreme ONE OS commands from the Linux shell.

3. To exit and return to the default Extreme ONE OS Linux shell, enter **exit**.

```
[root@device]# exit
exit
[admin@device]#
```

4. To exit the default Extreme ONE OS Linux shell and return to the Extreme ONE OS CLI, enter **exit**.

```
[admin@device]# exit
exit
device#
```

Saving and Appending Show Command Output to a File

1. Save the **show** command output to a file.

```
device# show running-config system aaa authentication | save auth_status
device#
```

In this example, the **show running-config system aaa authentication** output is saved to the auth-status file.

2. Append the show output to an existing file

```
device# show interface brief | append auth_status
```

The **show interface brief** command output is appended to the auth_status file.

Logs of Linux Shell Activities

By default, Extreme ONE OS logs users entering the ONE OS Linux shell, commands executed in that shell, and users exiting from the Linux shell back to the Extreme ONE OS CLI.

Firmware Fullinstall Support

This topic outlines the step-by-step process for performing a full, clean install of the provided Extreme ONE OS firmware. The process safely removes existing partitions while preserving essential SSH and certification files.



Note

Ensure that you have a valid URL containing the appropriate ExtremeONEOS binary image.

Key Features

- Complete clean firmware installation using the Extreme ONE OS CLI
- Managed partition deletion and creation
- Replacement of current configuration with default Extreme ONE OS settings
- Automatic preservation of SSH certification files

Event Log Messages

Event Type	Log Message
Pre-install event	Preparing Device for Fullinstall...This will take some time
Invalid image extension event	Fullinstall failed. Cannot be done on .app and .incr images
Successful validation event	Device ready for Fullinstall in %s. Device will reboot now.

Important Extreme ONE OS Configuration and Certificate Changes

Extreme ONE OS Configurations: Installing new firmware will overwrite your current configuration with the default Extreme ONE OS settings. To preserve your existing configuration, back it up to an external server using **copy running-config file <external server details>** command. You can then restore it using **copy file <external server details> running-config** command.

Management Certificates: The following folders will be automatically backed up and restored during the firmware installation. If you want to remove these certificate files, use the 'no-preserve' CLI option.

- /etc/ssh
- /var/data/cert-mgmt/ca-trust

- /var/data/cert-mgmt/app-cert
- /var/data/cert-mgmt/jwt
- /var/data/ztp

CLI Commands

The following table describes CLI commands to complete the system firmware fullinstall:

Full Syntax	system firmware fullinstall <url path of firmware file>		
Parameter descriptions	Parameter	Type	Description
	URL	string	URL or Filepath of firmware. disk://firmware/<filename> -OR- usb://<filename> -OR- scp[sftp]://<username>:<password>@<host>[:port]/<filepath> [vrf vrf-name] -OR- http[https]://[username:password@]<host>[:port]/<filepath> [vrf vrf-name]
	no-preserve	optional argument	By default, it's preserved or optionally users can prefer not to preserve using optional parameter no-preserve. wherever applicable, modify this.
Command mode	exec mode		
Permissions & Validations	Admin user only		
Behavior description	<p>This command simulates the 'onie-nos-install' process for new Extreme ONE OS firmware. Here's what it does:</p> <ul style="list-style-type: none"> • Deletes all device partitions • Preserves management certification files by default (unless 'no-preserve' option is used) • Initiates new ExtremeONE OS firmware installation from ONIE • Restores SSH configuration file and necessary certification files" 		

Help strings	<pre> 32d# system firmware commit Commit firmware version fullinstall Full Install firmware on switch rollback firwmare version uninstall Uninstall App update Update firmware on switch 32d# system firmware fullinstall URL disk://firmware/<filename> -OR- usb://<filename> -OR- scp[sftp]:// <username>:<password>@<host>[:port]/<filepath> [vrf vrf-name] -OR- http[https]: //[username:password@]<host>[:port]/<filepath> [vrf vrf-name] 32d# system firmware fullinstall <image path> # system firmware fullinstall dis <cr> no-preserve Remove management certificates </pre>
Error messages	<pre> Fullinstall failed. Cannot be done on .app and .incr images. Firmware image validation failed. Invalid extension for <wrong image path> %Error: Vrf is applicable only for scp[sftp] and http[https]. </pre>
Related commands	show version, show firmware

Related PROTO files	<p>The following openconfig gNOI RPC are used or supported:</p> <pre>rpc Activate(ActivateRequest) returns (ActivateResponse); rpc Verify(VerifyRequest) returns (VerifyResponse);</pre> <p>The following are the corresponding Request and Response messages:</p> <pre>// The ActivateRequest is sent by the Client to the // Target to initiate a change // in the next bootable OS version that is to be // used on the Target. // Dual Supervisor Target which requires installing // the entire system with // one Install RPC MUST return // NOT_SUPPORTED_ON_BACKUP error when requested // To Activate on standby Supervisor. message ActivateRequest { // The version that is required to be activated // and optionally immediattely // booted. string version = 1; // For dual Supervisors setting this flag // instructs the Target to perform the // action on the Standby Supervisor. bool standby_supervisor = 2; // If set to 'False' the Target will initiate the // reboot process immediattely // after changing the next bootable OS version. // If set to 'True' a separate action to reboot // the Target and start using // the activated OS version is required. This // action CAN be executing // the gNOI.system.Reboot() RPC. bool no_reboot = 3; } // The ActivateResponse is sent from the Target to // the Client in response to the // Activate RPC. It indicates the success of making // the OS package version // active. message ActivateResponse { oneof response { ActivateOK activate_ok = 1; ActivateError activate_error = 2; } } // If the Target is already running the requested // version in ActivateRequest, // then it replies with ActivateOK. If the Target // has the OS package version // requested in ActivateRequest then it replies with // ActivateOK and proceeds to // boot. // A dual Supervisor Target which requires // installing the entire system with // one Install RPC, will activate the image on all // Supervisors in response to // one Activate RPC. The Target should activate the // image on both Supervisors // with the least impact possible to forwarding. //</pre>
----------------------------	--

	<pre> // On a dual Supervisor Target which requires one Install RPC per supervisor, // performing this RPC on the Active Supervisor triggers a switchover before // booting the (old)Active Supervisor. The Target should perform a switchover // with the least impact possible to forwarding. message ActivateOK { } message ActivateError { enum Type { // An unspecified error. Must use the detail value to describe the issue. UNSPECIFIED = 0; // There is no OS package with the version requested for activation. This is // also used for an empty version string. NON_EXISTENT_VERSION = 1; // Dual Supervisor Target which requires installing the entire system // with one Install RPC MUST return NOT_SUPPORTED_ON_BACKUP error when // requested to Activate on standby Supervisor. NOT_SUPPORTED_ON_BACKUP = 2; } Type type = 1; string detail = 2; } message VerifyRequest { } message VerifyResponse { // The OS version currently running. string version = 1; // Informational message describing fail details of the last boot. This MUST // be set when a newly transferred OS fails to boot and the system falls back // to the previously running OS version. It MUST be cleared whenever the // systems successfully boots the activated OS version. string activation_fail_message = 2; VerifyStandby verify_standby = 3; // Dual Supervisor Targets that require the Install/Activate/Verify process // executed once per supervisor reply with individual_supervisor_install set // to true bool individual_supervisor_install = 4; } </pre>
Example output using gNOI	<pre> \$ gnoic -a 192.x.x.x:4x3 -u admin -p <default password> --tls-ca ca.cert.pem os activate --version fullinstall-/var/data/disk/ firmware/ExtremeOneSR-22.2.0.0.bin </pre>



TPVM IAH Extension

[About Integrated Appliance Hosting \(IAH\)](#) on page 60

[Configure TPVM](#) on page 61

[Config CLI](#) on page 62

[Exec CLI or gNOI](#) on page 68

[Show or gNOI CLI](#) on page 70

[RASlog Errors or Logs](#) on page 72

Use this topic to learn about the TPVM IAH Extension.

About Integrated Appliance Hosting (IAH)

IAH is a microservice in the Extreme ONE OS operating environment that manages the deployment and lifecycle of virtual appliances (preconfigured VMs). It supports open, non-proprietary VMs, allowing customers to run third-party workloads directly on the switch. IAH is built on a microservices architecture and utilizes KVM (Kernel-based Virtual Machine) for full virtualization.

Key features include:

- Deploy and manage virtual appliances
- Support for fully open VMs
- Run third-party VM-based workloads on the switch
- Eliminates the need for dedicated servers in data centers
- Leverages Qemu for machine emulation and virtualization
- Libvirt API for virtualization management
- Libvirtd daemon for managing VM activities
- Virsh command-line interface for VM management
- Passthrough mechanism for assigning physical devices to VMs

You can use IAH in the following scenario:

- Data analytics applications
- Traffic monitoring applications
- Network troubleshooting applications
- EFA and XCO applications running as VMs
- Other customer-specific applications

Configure TPVM

To utilize TPVM features and configuration CLIs, ensure the following:

- Set VM Name: The virtual machine name must be set to "TPVM".
- Copy Image: Copy the TPVM image to disk://iah/ using SCP before starting the deployment process.

```
device# copy file scp://xxx@<ip-addr>:/tmp/tpvm-4.7.6-0.amd64.deb file disk://iah/
```

Follow this procedure to configure TPVM.

For details on command syntax and parameters for TPVM configuration and management, see *Extreme ONE OS Switching v22.2.0.0 Command Reference Guide*.

1. Download TPVM software.

```
# copy file <scp_url> file disk://iah/tpvm-4.7.1-0.amd64.deb
```

2. Download the LDAP CA CERT files for the system.

```
# copy file <scp_url> file LDAP_CA_CERT_FILE
```

3. Create a default IAH VM configuration named "TPVM" with CPU:=4, RAM:=8GB, and DISKS.

```
# iah import TPVM tpvm disk://iah/tpvm-4.7.1-0.amd64.deb
```

4. Verify VM configuration.

```
# show iah vm TPVM
```

5. Switch to configuration mode.

```
# conf terminal
```

6. Access IAH configuration mode.

```
device(config)# iah
```

7. Configure the IAH TPVM configuration block.

```
device(config-iah)# vm TPVM
```

8. Example configuration to update CPU settings.

```
device(config-iah-vm-TPVM)# cpu 4
```

9. Example configuration to update RAM settings.

```
device(config-iah-vm-TPVM)# ram 6000
```

10. Set the hostname.

```
device(config-iah-vm-TPVM)# hostname ganga
```

11. Set the timezone.

```
device(config-iah-vm-TPVM)# timezone Africa/Casablanca
```

12. Set the password (default user: TPVM, sudo user: extreme).

```
device(config-iah-vm-TPVM)# password myPassPhrase
```

13. Configure any other supported attributes.

```
device(config-iah-vm-TPVM)# ...
```

14. Activate TPVM management network (can be done pre- or post-deployment).

```
device(config-iah-vm-TPVM)# activate-network
```

15. Example configuration for editing disk settings.

```
device(config-iah-vm-TPVM)# disk ...
```

16. Deploy the configuration.

```
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# deploy
```

17. Set IPv4 address and gateway.

```
device(config-iah-vm-TPVM)# interface management ipv4 2.2.2.2/24 gw 2.2.2.1
```

18. Change the hostname.

```
device(config-iah-vm-TPVM)# hostname tpmvmdut
```

19. Add/update NTP servers.

```
device(config-iah-vm-TPVM)# ntp 20.21.22.23
```

20. Update password.

```
device(config-iah-vm-TPVM)# password $6$abcd$fewofnawfnoawfnqof
```

21. Access TPVM console.

```
device# iah vm console TPVM
```

a. Shutdown TPVM.

```
device# iah vm shutdown TPVM
```

b. Start TPVM.

```
device# iah vm start TPVM
```

c. Reboot TPVM

```
device# iah vm reboot TPVM
```

Config CLI

Use this topic to learn about comprehensive overview of Config CLI and gNMI examples for the TPVM feature in IAH.

ONE OS IAH Equivalence

1. IAH allows configuration for certain attributes. For extended TPVM support, additional configurable attributes are available under the sub container "tpvm".

Config Mode – There are two configuration modes. But ONE OS IAH **Console CLI** hides this complexity for better user experience. Whereas the gNMI client must use the correct keypath.

The following table shows the command list structured according to the configuration mode and key path:

IAH Config Attributes (for any vm Name)	IAH Config TPVM Extension Attributes (only for vm Name == TPVM)
a. vm name b. vm description c. cpus d. ram e. autoboot f. pwless (<i>implicit</i>) g. active-network h. network :: mac i. enable j. disks :: disk k. insight / internal	a. hostname b. timezone c. password d. ethifs :: dhcp ipv4 address gw address e. dns f. ntp (up to 5) g. ldap h. ldap ca-cert i. trusted-peer

2. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf terminal
ONEOS(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)#
device# conf ter
device(config)# iah

device(config-iah)# no vm TPVM
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/config/name::string::TPVM
no form:
/system/iah/vms/vm[name=TPVM]/config/name
```

3. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# deploy

device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no deploy
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/config/deploy::bool::true
no form:
/system/iah/vms/vm[name=TPVM]/config/deploy::bool::false
```

4. Not Applicable. The following information should be sufficient:

```
device(config-iah-vm-TPVM)# no deploy      (or)
device(config-iah)# no vm TPVM
```

5. Not Applicable. ONE OS will also provide implicit password-less access for "extreme@TPVM".

6. Implicit for ONE OS. ONE OS has this configuration in place, rendering it unnecessary for XCO.



Note

To ensure all ports are properly initialized and configured, ONE OS restarts TPVM on every reboot due to libvirtd.

- IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# auto-boot

device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no auto-boot
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/config/auto-boot:::bool:::true

no form:
/system/iah/vms/vm[name=TPVM]/config/auto-boot:::bool:::false
```

7. When deploying TPVM for XCO, the /apps directory will be automatically created and mounted. If a specific size is required, users can utilize the following method:

- IAH Config Attributes (for any vm Name)

- Console cmd

```
device# show running-config iah
iah
  vm TPVM
    cpus 1
    ram 1024
    disk sda
      size 5000 [size is optional, derived from file if
file]
      format qcow2 [format is optional, qcow2 is default]
      bus virtio [bus is optional, virtio is default]
      file efaApps.qcow2 [preferred file name only]
    !
  !
!
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/disks/disk[name=sda]/config/name:::string:::sda
/system/iah/vms/vm[name=TPVM]/disks/disk[name=sda]/config/size:::uint32:::5000
/system/iah/vms/vm[name=TPVM]/disks/disk[name=sda]/config/file-format:::
string:::qcow2
/system/iah/vms/vm[name=TPVM]/disks/disk[name=sda]/config/bus-
type:::string:::virtio/system/iah/vms/vm[name=TPVM]/disks/disk[name=sda]/config/
source-file:::string:::efaApps.qcow2
```

8. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
```



```
device(config-iah-vm-TPVM)# activate-network
NO form is not applicable for this configuration
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/config/activate-network:::bool:::true
no form: NA
```

9. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# password myPassPhrase

device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no password
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/config/password:::string:::myPassPhraseno form:
/system/iah/vms/vm[name=TPVM]/tpvm/config/password
```

10. IAH Syntax:

```
[no] interface management ipv4 { dhcp | {<ipv4>/m [gw <ipv4>]} }
[no] interface management ipv6 { dhcp | {<ipv6>/m [gw <ipv6>]} }
```

IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# interface management ipv4 2.2.2.2/24
or
device(config-iah-vm-TPVM)# interface management ipv4 2.2.2.2/24 gw 2.2.2.1
or
device(config-iah-vm-TPVM)# interface management ipv4 dhcp
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah)# no interface management
device(config-iah-vm-TPVM)# no interface management ipv4
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-addr:::string:::2.2.2.2/24
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-gw:::string:::2.2.2.1
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/dhcp4:::bool:::true
no form:
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-addr/system/iah/vms/
vm[name=TPVM]/tpvm/management/config/ipv4-gw
```

11. Not Applicable

- ONE OS Switching – RME uses insight/internal port, so unavailable for other insight/internal purposes.

12. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# hostname ganga

device# conf ter
```

```
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no hostname
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/config/hostname:::string:::ganga
no form:
/system/iah/vms/vm[name=TPVM]/tpvm/config/hostname
```

13. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# timezone Africa/Casablanca
device# conf ter

device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no timezone
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-addr:::string:::2.2.2.2/24
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-gw:::string:::2.2.2.1
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/dhcp4:::bool:::true
no form:
/system/iah/vms/vm[name=TPVM]/tpvm/management/config/ipv4-addr/system/iah/vms/
vm[name=TPVM]/tpvm/management/config/ipv4-gw
```

14. IAH

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# dns primary-server 8.8.8.8 secondary-server 9.9.9.9
domain example.com
device(config-iah-vm-TPVM)# dns primary-server 8.8.8.8 secondary-server 9.9.9.9
device(config-iah-vm-TPVM)# dns primary-server 8.8.8.8
/* every set is a fresh set */
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no dns - this will purge entire configuration
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/dns/config/primary-server:::string:::8.8.8.8/
system/iah/vms/vm[name=TPVM]/tpvm/dns/config/secondary-server:::string:::9.9.9.9/
system/iah/vms/vm[name=TPVM]/tpvm/dns/config/domain:::string:::example.com/*
/system/iah/vms/vm[name=TPVM]/tpvm/dns/config/apply */no form:
/system/iah/vms/vm[name=TPVM]/tpvm/dns
/system/iah/vms/vm[name=TPVM]/tpvm/dns/config/primary-server:::string:::8.8.8.8/
system/iah/vms/vm[name=TPVM]/tpvm/dns/config/secondary-server:::string:::9.9.9.9/
system/iah/vms/vm[name=TPVM]/tpvm/dns/config/domain:::string:::example.com
```

15. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# ntp 1.1.1.1
device(config-iah-vm-TPVM)# ntp 2.2.2.2
device(config-iah-vm-TPVM)# ntp 3.3.3.3
device(config-iah-vm-TPVM)# ntp time.example.com
device(config-iah-vm-TPVM)# ntp 2001:db8::1
```

```

device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no ntp 3.3.3.3
device(config-iah-vm-TPVM)# no ntp          - this will purge entire configuration

```

- gNMI keypath

```

/system/iah/vms/vm[name=TPVM]/tpvm/ntp/servers[server=1.1.1.1]/config/
server:::string:::1.1.1.1/system/iah/vms/vm[name=TPVM]/tpvm/ntp/
servers[server=2.2.2.2]/config/server:::string:::2.2.2.2/system/iah/vms/
vm[name=TPVM]/tpvm/ntp/servers[server=time.example.com]/config/
server:::string:::time.example.com/system/iah/vms/vm[name=TPVM]/tpvm/ntp/
servers[server=2001:db8::1]/config/server:::string:::2001:db8::1no form:
/system/iah/vms/vm[name=TPVM]/tpvm/ntp/servers[server=3.3.3.3]/config/server/
system/iah/vms/vm[name=TPVM]/tpvm/ntp

```

16. TPVM LDAP Host

- Console cmd

```

device# conf ter
device(config)# iah
device(config-iah)# vm TPVM

device(config-iah-vm-TPVM)# ldap host 1.2.3.4 basedn dc=ldap,dc=hc-fusion,dc=in \
                                rootdn cn=admin,dc=ldap,dc=hc-fusion,dc=in rootdnpw
pwdstring

device(config-iah-vm-TPVM)# ldap host 1.2.3.4 port 4000 basedn dc=ldap,dc=hc-
fusion,dc=in \
                                rootdn cn=admin,dc=ldap,dc=hc-fusion,dc=in rootdnpw
pwdstring
device(config-iah-vm-TPVM)# ldap host 1.2.3.4 port 4000 secure basedn dc=ldap,dc=hc-
fusion,dc=in \
                                rootdn cn=admin,dc=ldap,dc=hc-fusion,dc=in rootdnpw
pwdstring
/* every set is a fresh set */
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no ldap

```

- gNMI keypath

```

/system/iah/vms/vm[name=TPVM]/tpvm/ldap/config/host:::string:::8.8.8.8/
system/iah/vms/vm[name=TPVM]/tpvm/ldap/config/port:::uint16:::4000/system/iah/vms/
vm[name=TPVM]/tpvm/ldap/config/secure:::bool:::true/system/iah/vms/vm[name=TPVM]/
tpvm/ldap/config/basedn:::string:::dc=ldap,dc=hc-fusion,dc=in/system/iah/vms/
vm[name=TPVM]/tpvm/ldap/config/rootdn:::string:::cn=admin,dc=ldap,dc=hc-
fusion,dc=in/system/iah/vms/vm[name=TPVM]/tpvm/ldap/config/
password:::string:::pwdstringno form:
/system/iah/vms/vm[name=TPVM]/tpvm/ldap/system/iah/vms/vm[name=TPVM]/tpvm/ldap/
config/basedn:::string:::dc=ldap,dc=hc-fusion,dc=in/system/iah/vms/vm[name=TPVM]/
tpvm/ldap/config/rootdn:::string:::cn=admin,dc=ldap,dc=hc-fusion,dc=in/
system/iah/vms/vm[name=TPVM]/tpvm/ldap/config/password:::string:::pwdstringno form:
/system/iah/vms/vm[name=TPVM]/tpvm/ldap

```

17. Download the TPVM LDAP CA certificates using SCP from a specified file location.

It is a **two steps process** as follows.

- Copy SCP URL to a file at host.

- b. Copy from host to TPVM (can be done directly).



Note

The config attribute can only be specified using the filename. The file copy operation can be performed using the COPY exec command. In the RMA use case, the CERT file can be made available on the host, and the filename can be used to place it on the TPVM.. Filename can be used to place it at TPVM.

18. IAH Config Attributes (for any vm Name)

- Console cmd

```
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM

device(config-iah-vm-TPVM)# trusted-peer ipv4 1.2.3.4 password peerPwdStr sudo-user
peerUsrName/* every set is a fresh set */
device# conf ter
device(config)# iah
device(config-iah)# vm TPVM
device(config-iah-vm-TPVM)# no trusted-peer
```

- gNMI keypath

```
/system/iah/vms/vm[name=TPVM]/tpvm/trusted-peer/config/peer-
ip:::string:::8.8.8.8/system/iah/vms/vm[name=TPVM]/tpvm/trusted-peer/config/peer-
password:::string:::peerPwdStr/system/iah/vms/vm[name=TPVM]/tpvm/trusted-peer/
config/peer-sudoer:::bool:::peerUsrNameno form:
/system/iah/vms/vm[name=TPVM]/tpvm/trusted-peer
```

Exec CLI or gNOI

Use this topic to learn about comprehensive overview of Exec CLI and gNOI examples for the TPVM feature in ONE OS IAH.

ONE OS IAH Equivalence

- Two step process:

Console CLI	gRPC
Step 1: Download TPVM image. Syntax: copy file <URL> file <URL> [source-ip ADDRESS] [vrf VRF-NAME] device# copy file scp://fvt:pray4green@192.x.x.x:/ / buildsjc/sre_fusion/Nightly/tpvm/ tpvm4.6.17/LATEST_BUILD/dist/ SWBD2900/tpvm-4.7.1-0.amd64.deb file disk://iah/ tpvm-4.7.1-0.amd64.deb vrf mgmt-vrf	gRPC: <pre> service ConfigManagement { ... rpc CopyRunningToFile(CopyConfigRequest) returns (ConfigResponse) {}; ... }</pre>
Note: Src URL can be only scp/usb/file.	
Step 2: Deploy new to upgrade. Syntax: iah vm <NAME> upgrade file <filename> [incremental snapshot] device# iah vm TPVM upgrade file tpvm-4.7.1-0.amd64.deb device# iah vm TPVM upgrade file tpvm-4.7.1-0.amd64.deb \ snapshot device# iah vm TPVM upgrade file \ tpvm_inc_upg-4.6.17-1.amd64.deb incremental	<pre> rpc TpvmUpgrade(TpvmUpgradeReq) returns (TpvmUpgradeResp) {};</pre>

- IAH may have multiple PTM files. Displays all the PTM files by default, or displays specified file if argument is given.

Console CLI	gRPC
Syntax: iah fileinfo IMAGE_FILENAME IMAGE_FILENAME := disk://iah/ FILENAME FILENAME := <any_valid_filename_string> For TPVM debian file, this command validates the available file, filename, size, and version using Debian Meta. Prefix disk://iah optional. When absolute path is not provided, it will look in the disk://iah folder.	gRPC: <pre> rpc IahFileInfo(IahFileInfoReq) returns (IahFileInfoResp) {};</pre>

Console CLI	gRPC
Syntax iah vm TPVM snapshot create iah vm TPVM snapshot create withhefa iah vm TPVM snapshot revert iah vm TPVM snapshot delete	gRPC: updaterpc VmBackup(VmkBackupReq) returns (VmBackupResp) {};

4. **syntax:** iah vm < start | shutdown | poweroff | reboot | suspend | resume
> <NAME>

Console CLI	gRPC
Syntax: iah vm <NAME> start iah vm <NAME> stop	gRPC VmControlCommand(VmControlcommandReq) returns (VmControlCommandResp) {}

5. **Syntax:** iah vm <NAME> console
iah vm TPVM console
6. Not Applicable (use COPY command)
7. Not Applicable. Refer "no vm Name" or "no deploy".
8. This is an Exec CLI. It is prerequisite step in IAH, which assist to prefill few Computing Defaults Config like CPU/RAM/DISK, before "deploy".This allows you to review these defaults and if required alter them before the deployment.

Console CLI	gRPC
Syntax: iah vm TPVM import <ova/tpvm> <FILE> device(config)# iah import TPVM tpvm disk://iah/ tpvm-4.7.1-0.amd64.deb	gRPC: rpc VmImport(VmImportReq) returns (VmImportResp) {}

Show or gNOI CLI

Use this topic to learn about a summary of show or gNOI CLI examples for the TPVM feature in IAH for ONE OS.

ONE OS IAH Equivalence

1. **Syntax:** **show running-config iah**

Example: **show running-config iah**

Console CLI

- Sample output

```
device# show running-config iah
iah
  vm Ubuntu1
    description Ubuntu demo vm
    Auto-boot true
    activate-network
```

```

cpus 1
ram 1024
disk sda
    size 1000
    format qcow2
    bus sata
    file disk://iah/disk1.qcow2
!
network vmmgmt
    interface management 0
    mac 11:22:33:44:55:00
!
!
!
```

- Fetch Config Data

The running configuration retrieves data from the Config DB (CDB), so the key path used is same to the one used during initial configuration.

```
/system/iah/vms/vm[name=TPVM]/...
```

Example: Get Time Zone - configured data

```
/system/iah/vms/vm[name=TPVM]/tpvm/config/timezone
```

2. Console CLI

- Example output

```

device# show iah vm all
Number of virtual machines: 2
Name: centos1
    id: 5
    deploy-status: DOWNLOADING_IMAGE
    status:
    error:
    description: centos vm
    autostart: disabled
    cpus: 2
    ram: 2048 MB
    disk: sda
        size: 1000 MB
        file: flash://iah/centos-vm.qcow2
        file format: QCOW2
        bus type: VIRTIO
Name: TPVM
    id: 4
    deploy-status: DEPLOY_SUCCESSFUL
    status: RUNNING
    error:
    description: demo ubuntu vm
    autostart: enabled
    cpus: 1
    ram: 1024 MiB
    disk: sda
        size: 5000 MB
        file: flash://iah/ubuntu-disk1.qcow2
        file format: QCOW2
        bus type: SATA
```

- Fetch Operational Data

Operational data for each configurable attribute is retrieved from the State DB (SDB), using the same key path as initial config, but with the 'state' keyword instead.

```
/system/iah/vms/vm[name=TPVM] /...
```

Example: Get Time Zone - operational data

```
/system/iah/vms/vm[name=TPVM] /tpvm/state/timezone
```

- List of important non-config attributes

- AH VM - deploy-status

```
/system/iah/vms/vm[name=TPVM] /state/deploy-status
```

- AH VM - vmstatus

```
/system/iah/vms/vm[name=TPVM] /state/vm-status
```

- IAH VM Last Oper Error - error

```
/system/iah/vms/vm[name=TPVM] /state/error
```

- TPVM Version

```
/system/iah/vms/vm[name=TPVM] /tpvm/state/version
```

- Snapshot

```
/system/iah/vms/vm[name=TPVM] /snapshot/state/version /
system/iah/vms/vm[name=TPVM] /snapshot/state/time /system/iah/vms/
vm[name=TPVM] /snapshot/state/size /system/iah/vms/vm[name=TPVM] /
snapshot/state/description
```

RASlog Errors or Logs

Use this topic to learn about the CLI examples for the TPVM feature in IAH for ONE OS.

ONE OS IAH Equivalence

The IAH RASlogs is simplified to focus on key events, such as Operation start, Operation completion, and Operation failure (with reason and error details).

This approach provides following benefits:

- XCO doesn't need to integrate new RASlog IDs
- Fixed string fields for operation names
- Evolving reason fields for detailed error information

The proposed RASlog IDs are:

- 27037: Operation started (VM name, operation, bid)
- 27038: Operation completed (VM name, operation, bid)
- 7039: Operation failed (VM name, operation, bid, VM status, reason)

The bid (Internal Transaction Counter ID) helps track transactions and matches them with SupportSave data and timestamps.

IAH uses ONE OS COPY command for TPVM download related RASlogs.



Network Time Protocol (NTP)

[Network Time Protocol overview](#) on page 74

[Configuring NTP](#) on page 76

[Displaying the NTP Server Status](#) on page 77

Network Time Protocol overview

Network Time Protocol (NTP) maintains uniform time across all devices in a network. The NTP commands support the configuration of an external time server to maintain synchronization among all local clocks in a network.

To keep the time in your network current, it is recommended that each device have its time synchronized with at least one external NTP server.

Date and Time Settings

Extreme devices maintain the current date and time inside a battery-backed real-time clock (RTC) circuit. Date and time are used for logging events. Device operation does not depend on the date and time; a device with incorrect date and time settings can function correctly. However, because the date and time are used for logging, error detection, and troubleshooting, you should set them correctly.

Time Zone Settings

The time zone settings have the following characteristics:

- The setting automatically adjusts for Daylight Savings Time.
- Changing the time zone on a device updates the local time zone setup and is reflected in local time calculations.
- Default NTP configuration is not present.
- System services that have already started will reflect the time zone changes only after the next reboot.
- Time zone settings persist across failover for high availability.
- Time zone settings are not affected by NTP server synchronization.

Network Time Protocol Server Overview

The Network Time Protocol server is used to obtain the correct time from an external time source and adjust the local time in each connected device. When NTP server functionality is enabled, the NTP server starts listening on the NTP port for client requests and responds with the reference time. Up to eight server addresses can be configured in IPv4 or IPv6 format. When multiple NTP server addresses are configured, the NTP algorithm finds the most reliable server and uses this as the active NTP server. If there are no reachable time servers, then the local device time becomes the default time until a new active time server is configured. If an NTP server loses synchronization, it will operate in master mode to serve time using the local clock.

The NTP server is stateless and does not maintain NTP client information. Network time synchronization is guaranteed only when a common external time server is used by all devices.



Important

Although time-stepping corrects a large offset after a reload, as a best practice do not manually change the time after NTP synchronization.

Network Time Protocol Client Overview

The NTP client maintains the server and peer state information as an association. The server and peer association is mobilized at startup, or after it has been configured. A statically configured server/peer association is not demobilized unless the configuration is removed/changed. A symmetric passive association is mobilized upon the arrival of an NTP packet from a peer which is not statically configured. This type of association is demobilized on error or timeout.

The NTP client operation can be summarized as follows:

1. The device is booted and the system initializes. The configured servers and peers are polled at the configured poll interval. Additional dynamically discovered servers/peers are also polled.
2. Multiple samples of server/peer times in the NTP packet are added to and maintained in the association database.
3. The selection, cluster, and combine algorithms choose the most accurate and reliable server/peer as system peer.



Note

Refer to RFC 5905.

4. The reference time from the system peer is used for system time synchronization.
5. The NTP client increases the poll interval from the minimum poll interval to the maximum poll interval value after the clock stabilizes.

After the system peer is chosen, the system time is synchronized using one of the following ways:

- If the system time differs from the system peer by less than 128 milliseconds, then the system clock is adjusted slowly towards the system peer time reference time.

- If the system time differs from the system peer by greater than 128 milliseconds, then the system clock is stepped to the system peer reference time. The old, time-related information stored in the server/peer association database is cleared.

Network Time Protocol Associations

The following modes are the NTP polling based associations:

1. NTP server
2. NTP client
3. NTP peer

NTP Server

The Server mode requires no prior client configuration; it responds to Client mode NTP packets. The **enable** command is used to set the device to operate in Server mode. Use **no enable** to ensure NTP is configured in server mode.

NTP Client

When the system is operating in Client mode, all configured NTP servers and peers are polled. The device selects a host from all the polled NTP Servers from which to synchronize. To configure the NTP servers and peers individually, use the **server** and **peer** commands.

NTP Peer

NTP Peer mode is intended for configurations where a group of devices operate as mutual backup for one another. If one device loses a reference source, the time values flow from the remaining peers.

Configuring NTP

The date and time are set in privileged EXEC mode and only have to be configured once per device because the value is written to nonvolatile memory. After the basic time information is set up, an NTP server is configured to allow the local time to be synchronized across the network.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Synchronize the local time with an external source.

```
device(config)# system
device(config-system)# ntp
device(config-system-ntp)# enable
device(config-system-ntp)# vrf mgmt-vrf
device(config-system-ntp)# server 2.2.2.2
device(config-system-ntp)# peer 3.3.3.3
device(config-system-ntp)#
```

3. Exit to privileged EXEC mode.

```
device(config-system-ntp)# exit
device(config-system)# exit
device(config)# exit
```

4. Display the active NTP server IP address.

```
//Before enabling

device# show ntp status
Clock is unsynchronized, no reference clock
NTP client mode is disabled

//After time sync-up

device# show ntp status
Clock is synchronized, stratum 3, reference clock is 5.xx.xx.xx,
precision is -23,
reference time is e807413e.bflabdc2 Thu, May 11 2023 10:26:06.746,
clock offset is +0.000000, root delay is 74.781,
root dispersion is 190.533, peer dispersion is 20579,
NTP client mode is enabled
```

Displaying the NTP Server Status

You can verify the NTP server status. When an NTP server has been configured, the server IP address is displayed. If an NTP server is not configured or the server is unreachable, the output displays LOCL (for local device time).

NTP Server Status When an NTP Server is Not Configured

When an NTP server is not configured, the device will work with local time and the NTP status will display as shown below:

```
device# show ntp status
Clock is unsynchronized, no reference clock
NTP client mode is disabled

device#
```

NTP Server Status When an NTP Server is Configured

The following example shows the status of a configured NTP server:

```
device(config-system-ntp)# show ntp status
Clock is synchronized, stratum 3, reference clock is 1x.xx.xx.xxx,
precision is -23,
reference time is eb687879.fff75557 Tue, Feb 25 2025 9:12:25.999,
clock offset is -0.657594, root delay is 68.706,
root dispersion is 104.294, peer dispersion is 21735,
NTP client mode is enabled (VRF: mgmt-vrf)

device(config-system-ntp)#
```



SNMP

[SNMP Overview](#) on page 78
[Configuring SNMPv2](#) on page 81
[Configuring SNMPv3](#) on page 82
[Supported MIBs](#) on page 83

SNMP Overview

Simple Network Management Protocol (SNMP) is a set of application layer protocols for exchanging management information between network devices.

SNMP enables network administrators to monitor and manage devices on a network by sending and receiving messages, known as protocol data units (PDUs). These messages, called SNMP Get-Requests, allow administrators to track specific data values.

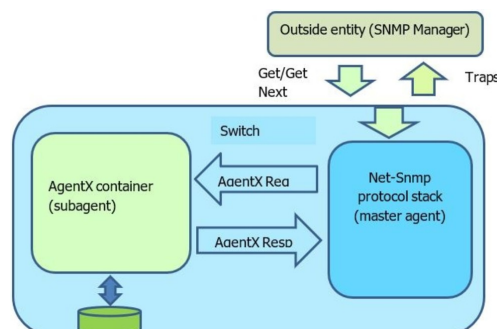
The SNMP server on the Extreme Networks device supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

Network administrators can retrieve data from devices in the following two ways using Get/GetNext requests, supporting both IPv4 and IPv6 device addresses:

1. SNMP v1/v2c: Using a community string to authenticate.
2. SNMP v3: Using a username, authentication password (MD5/SHA), and encryption password (AES/DES) for secure authentication and data encryption.

SNMP Traps are alert messages sent from SNMP-enabled devices to a central collector (SNMP manager), notifying it of significant issues and events. In Extreme ONE OS, SNMP support is provided through the open-source 'net-snmp' protocol stack.

The 'agentx subagent' container collects required MIB data from the state database (and other feature services if needed) and feeds it to the 'net-snmp' protocol stack.



Network management using SNMP requires three components:

- **SNMP manager**—Typically, network management systems (NMS) manage networks by monitoring the network parameters, and optionally, setting parameters in managed devices. The SNMP manager communicates to the devices within a network using the SNMP protocol.
- **SNMP agent**—Software that resides in the managed devices in the network, and collects and stores data from these devices. Each device hosts an SNMP agent. The agent receives requests from the SNMP manager and responds with the requested data. In addition, the agent can asynchronously alert the SNMP manager about events by using special PDUs called traps.
- **Management Information Base (MIB)**—Hierarchical database where SNMP agents in the managed devices store the data about these devices. The MIB is structured on the standard specified in the RFC 2578 [Structure of Management Information Version 2 (SMIv2)].

An SNMP manager can issue read operations to retrieve and use the MIB objects to manage and monitor devices on the network. However, the MIB structure determines the scope of management access allowed by a device.

Support for Get/GetNext operations is provided for the following MIBs:

- IF-MIB: Interfaces (1.3.6.1.2.1.2.2), ifXTable (1.3.6.1.2.1.31.1.1)
- SYSTEM-MIB(RFC1213): (1.3.6.1.2.1.1)
- ENTITY-MIB: entPhysicalTable (1.3.6.1.2.1.47.1.1.1.1)

The device supports the configuration of trap hosts as a trap recipient to receive traps and optionally receive SNMP communication through a VRF. You can configure the SNMP trap receiver by specifying its IPv4/IPv6 address or hostname/domain name. Ensure to configure the SNMPv3 trap using the existing SNMPv3 user configuration. The following generic traps are supported:

- ColdStart
- LinkUp
- LinkDown



Important

- SNMP SET operation is not supported.
- Performing an SNMPwalk on all MIBs may result in the message 'No more variables left in this MIB View' indicating the end of the MIB tree has been reached.

Inband management

SNMP server can be configured to run on multiple VRF instances. SNMP trap hosts can be configured to send traps to hosts through specific VRF and specific source-interface.

Basic SNMP Operation

Every Extreme device carries an *agent* and management information base (MIB), as shown in the figure [Figure 1](#). The agent accesses information about a device and makes it available to an SNMP network management station.

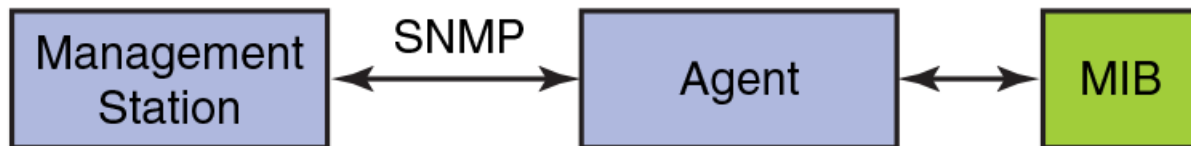


Figure 1: SNMP Structure

When active, the management station can "get" information when it queries an agent. SNMP commands, such as **get**, **getnext**, and **getresponse**, are sent from the management station, and the agent replies once the value is obtained or modified as shown in the figure [Figure 2](#). Agents use variables to report such data as the number of bytes and packets in and out of the device, or the number of broadcast messages sent and received. These variables are also known as managed objects. All managed objects are contained in a MIB.

The management station can also receive *traps*, unsolicited messages from the device agent if an unusual event occurs as shown in the next figure.



Figure 2: SNMP Trap

The agent can receive queries from one or more management stations and can send traps to up to six management stations.

SNMP Community Strings

You can use the community string to restrict the access of MIBs for SNMPv1 and SNMPv2c requests. You can configure a total of 256 read-only community strings on the device.

The software automatically hashes SNMP community strings.

By default, the community strings are displayed in hashed format in the running-config. However, you can configure the system to show them in plain text using the **unhide-secrets** CLI command.

SNMP Users

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services. This model provides a user that

is associated with security information for authentication of its generated SNMP messages.

SNMP Server Hosts

For an SNMPv3 trap, you associate a SNMPv3 host with the SNMP users. When you specify the host, you also specify a community string for SNMPv1 and SNMPv2. The Extreme device sends all the SNMP traps to the specified hosts and includes the specified community string. Then, administrators can filter for traps from a Extreme device based on IP address or community string.

SNMP Source Interface

The specified interface acts as the source interface for SNMP trap. SNMP trap host can be configured for SNMP version 1, version 2, and version 3 per instance. If the source interface is not specified, the source IP address is the IP address of the interface through which packet exits device. If the source interface is modified (changing IP address), then it is reflected in the trap packets. Configured source interface IP address is not cached because the corresponding IP address can be modified. While sending out the SNMP trap packets to find the source IP address to use, the system checks and picks up the source interface configured. If an interface with no IP address is configured as the source interface, SNMP trap packets have the egress interface IP as the source IP.

The SNMP source interface supports the following interface types:

- Virtual routing interface
- Loopback interface
- Ethernet
- Port Channel

Configuring SNMPv2

You can configure SNMP with user-defined VRF and source interface.

To configure SNMPv2, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

```
device# configure terminal
device(config)#
```

2. Add an SNMP community string.

```
device(config)# system
device(config-system)# snmp-server DEFAULT
device(config-system-snmp-server)# community public
device(config-system-snmp-server)#
```

This example adds the extremero community string to access the MIBs for SNMPv2c requests.

3. Configure the SNMP trap host associated with community string.

```
device(config-system-snmp-server)# vrf mgmt-vrf
device(config-system-snmp-server-vrf-mgmt-vrf)# enable
device(config-system-snmp-server-vrf-mgmt-vrf)# host 192.168.1.1 public
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-xxxx)# version 2c
```

```
device(config-system-snmp-server-vrf-rmgmt-vrf-host-192.168.1.1-xxxx)# udp-port 200
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-xxxx)# source-interface
ethernet 0/1
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-xxxx)
```

This example configures 192.x.x.x as a trap recipient with SNMPv2c on the target port 200 and associates the public community string.

4. Verify the configuration.

```
device(config-system-snmp-server)# do show running-config system snmp-server

system
 snmp-server DEFAULT
  vrf mgmt-vrf
  enable
  host 192.168.1.1 0ymX6XR7Za
  version 2c
  udp-port 200
  source-interface ethernet 0/1.0
  !
  !
  community-hashed 0ymX6XR7Za
  !
```

Configuring SNMPv3

SNMPv3 uses SNMP users to restrict SNMP access. When you map an SNMP user to an SNMP group, you can restrict the access of MIBs for SNMP requests through an SNMP view.

You can configure SNMP with user-defined VRF and source interface.

To configure SNMPv3, perform the following steps.

1. In privileged EXEC mode, enter global configuration mode.

```
device(config-system)#
```

2. Configure the SNMP server.

```
device(config-system)# snmp-server
device(config-system-snmp-server)#
```

Only an SNMP server instance named DEFAULT is allowed in this release.

3. Add an SNMP user.

```
device(config-system-snmp-server)#user user1 auth noauth
device(config-system-snmp-server)#user user2 auth md5 auth-password Authkey@12 priv
nopriv
device(config-system-snmp-server)#user user3 auth sha auth-password AuthKey@12 priv
aes priv-password User1@privkey
device(config-system-snmp-server)#user user4 auth md5 auth-password-hashed
0x9f1626506dbd7540c8ce526070fb69f9
device(config-system-snmp-server)#user user5 auth md5 auth-password-
hashed 0x9f1626506dbd7540c8ce526070fb69f9 priv aes priv-password-hashed
0x9f1626506dbd7540c8ce526070fb69f9
```

This example adds the user2 user to access of MIBs for SNMPv3 requests. For SNMPv3 users, the passwords for **auth-password** and **priv-password** keywords are encrypted while storing to the persistent memory or displaying it back to the user. You can configure either with a plain-text password or an encrypted password.

In both cases, the **show running-config** command displays the passwords as encrypted.

4. Configure the SNMPv3 trap host associated with an SNMP user.

```
device# configure terminal
device(config)# system
device(config-system)# snmp-server DEFAULT
device(config-system-snmp-server)# community extremero
device(config-system-snmp-server)# vrf mgmt-vrf
device(config-system-snmp-server-vrf-mgmt-vrf)# enable
device(config-system-snmp-server-vrf-mgmt-vrf)# host 192.168.1.1 user3
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-user3)# version 3
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-user3)# udp-port 200
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-user3)# source-interface
ethernet 0/1.0
device(config-system-snmp-server-vrf-mgmt-vrf-host-192.168.1.1-user3)
```

In this example, the trap host uses Ethernet interface 0/1.0 as the source interface, which determines the source IP address to be used while sending traps to the trap host:

5. Access privileged EXEC mode.

```
device(config-system)# exit
```

6. Verify the configuration.

```
device# show running-config system snmp-server
system
  snmp-server DEFAULT
    user user1 auth noauth
    user user2 auth md5 auth-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9 priv
  nopriv
    user user3 auth md5 auth-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9 priv
  aes priv-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9
    user user4 auth md5 auth-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9 priv
  nopriv
    user user5 auth md5 auth-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9 priv
  aes priv-password-hashed 0x9f1626506dbd7540c8ce526070fb69f9
    community extremero
    vrf mgmt-vrf
    enable
    host 192.168.1.1 user3
    version 3
    udp-port 200
    source-interface ethernet 0/1.0
  !
!
```

Supported MIBs

Extreme ONE OS supports the following MIBs and traps:



Note

For more details on supported MIBs and traps, refer to the *Extreme ONE Switching SNMP MIB Reference Guide*.

- Interface Group MIB
- System Group MIB
- Entity MIB

- ifXTable Extended MIB
- Supported Traps: Standard Traps, Enterprise MIB Traps



LLDP

[LLDP Overview](#) on page 85

[Configuring and Managing LLDP](#) on page 94

LLDP Overview

The IEEE 802.1AB Link Layer Discovery Protocol (LLDP) enhances the ability of network management tools to discover and maintain accurate network topologies and simplify LAN troubleshooting in multi-vendor environments. To efficiently and effectively operate the various devices in a LAN you must ensure the correct and valid configuration of the protocols and applications that are enabled on these devices. With Layer 2 networks expanding dramatically, it is difficult for a network administrator to statically monitor and configure each device in the network.

Using LLDP, network devices such as routers and switches advertise information about themselves to other network devices and store the information they discover. Details such as device configuration, device capabilities, and device identification are advertised. LLDP defines the following:

- A common set of advertisement messages.
- A protocol for transmitting the advertisements.
- A method for storing the information contained in received advertisements.



Note

- LLDP runs over the data-link layer which allows two devices running different network layer protocols to learn about each other.
- Link Layer Discovery Protocol (LLDP) is enabled globally by default.

LLDP information is transmitted periodically and stored for a finite period. Every time a device receives an LLDP advertisement frame, it stores the information and initializes a timer. If the timer reaches the time to live (TTL) value, the LLDP device deletes the stored information ensuring that only valid and current LLDP information is stored in network devices and is available to network management systems.

Layer 2 Topology Mapping

As LLDP devices transmit and receive advertisements, the devices store information they discover about their neighbors. Advertisement data such as a neighbor's

management address, device type, and port identification is useful in determining what neighboring devices are in the network.

**Note**

The Extreme Networks LLDP implementation supports up to 64 neighbors per interface.

The higher level management tools, such as the Network Advisor, can query the LLDP information to draw Layer 2 physical topologies. The management tools can continue to query a neighboring device through the device's management address provided in the LLDP information exchange. As this process is repeated, the complete Layer 2 topology is mapped.

In LLDP the link discovery is achieved through the exchange of link-level information between two link partners. The link-level information is refreshed periodically to reflect any dynamic changes in link-level parameters. The basic format for exchanging information in LLDP is in the form of a type, length, value (TLV) field.

LLDP keeps a database for both local and remote configurations. The LLDP standard currently supports three categories of TLVs. The Extreme Networks LLDP implementation adds a proprietary Extreme Networks extension TLV set. The TLV sets are described in the following topics:

Basic Management TLV Set

This set provides information to map the Layer 2 topology and includes the following TLVs.

Chassis ID TLV

Provides the ID for the switch or router where the port resides. Identifies the device transmitting the LLDP (Link Layer Discovery Protocol) frame. The Chassis ID TLV enables neighboring devices to uniquely identify the transmitting device, facilitating network discovery and management tasks. This is a mandatory TLV:

- Format
 - Type (7 bits): TLV type (typically 1 for Chassis ID)
 - Length (9 bits): Length of the TLV value field in bytes
 - Subtype (7 bits): Format or type of the Chassis ID
 - Chassis ID (Variable Length): Actual chassis identification data
- SubType Values
 - Chassis component type and number
 - Interface alias
 - Port component type and number
 - MAC address
 - Network address
 - Interface name

Port ID TLV

Provides a unique identifiable information of the port. The Port ID could be one of the following: MAC address, Network address, Interface name of the port. On the ONE OS, the interface name of the port is provided. The Port ID TLV enables neighboring devices to uniquely identify the transmitting port, facilitating network discovery, troubleshooting, and management tasks. This is a mandatory TLV.

This set provides information to map the Layer 2 topology and includes the following TLVs:

- Format
 - Type (7 bits): TLV type (typically 2 for Port ID)
 - Length (9 bits): Length of the TLV value field in bytes
 - Subtype (7 bits): Format or type of the Port ID
 - Port ID (Variable Length): Actual port identification data
- SubType Values
 - Interface alias
 - Port component type and number
 - MAC address
 - Network address
 - Interface name
 - Agent Circuit ID
 - Locally assigned value

Time to Live (TTL) TLV

Provides the remaining time for the information contained in the LLDPDU to be considered valid. The TTL TLV enables LLDP-enabled devices to manage the aging of received LLDP information, ensuring accurate network discovery and management by preventing stale information retention. This is a mandatory TLV.

- Format
 - Type (7 bits): TLV type (typically 3 for TTL TLV)
 - Length (9 bits): Length of the TLV value field in octets, typically 2 octets for TTL TLV
 - TTL (16 bits): Represents the time-to-live value in seconds, indicating the remaining time for the LLDPDU information to be considered valid

System Name TLV

Provides the system-assigned name in an alphanumeric format. The System Name TLV identifies the device transmitting the LLDP (Link Layer Discovery Protocol) frame. It provides a human-readable identifier for the device, simplifying network management, troubleshooting, and identification. This is an optional TLV.

- Format
 - Type (7 bits): Type (7 bits): TLV type (typically 2 for Port ID)
 - Length (9 bits): Length of the TLV value field (in octets)
 - System Name (Variable Length): Actual system name string

System Description TLV

Provides a description of the network entity in an alphanumeric format. This includes system name, hardware version, operating system, and supported networking software. It provides a textual description of a system, providing additional details such as model, manufacturer, firmware version, or capabilities. This TLV enables neighboring devices to access descriptive information about the transmitting device, facilitating network management, identification, and troubleshooting. This is an optional TLV.

- Format
 - Type (7 bits): Identifies the TLV type, typically set to 6 for System Description
 - Length (9 bits): Specifies the length of the TLV value field in bytes
 - System Description (Variable Length): Contains the actual system description string

Management Address TLV

Indicates the addresses of the local switch. Remote switches can use this address to obtain information related to the local switch. The Management Address TLV conveys network management addresses associated with the transmitting device, enabling remote management and monitoring. This TLV enables neighboring devices to obtain network management addresses, facilitating network management, monitoring, and troubleshooting. Additionally, you can configure the management address to be advertised via LLDP (Link Layer Discovery Protocol). This is an optional TLV.

- Format
 - Type (7 bits): Identifies the TLV type (typically 8 for Management Address TLV).
 - Length (9 bits): Specifies the TLV value field length in octets
 - Address Subtype (1 octet): Defines the management address type (e.g., IPv4, IPv6, MAC address)
 - Address Length (1 octet): Indicates the address data length in octets
 - Address (Variable Length): Contains the actual management address data

Port Description TLV

Provides a description of the port in an alphanumeric format. By providing a descriptive string, the Port Description TLV enables neighboring devices to obtain valuable information about the transmitting device's ports, facilitating network management and troubleshooting. This is an optional TLV.

- Format
 - Type (7 bits): Field indicating the TLV type. For Port Description TLV, the value is typically 4
 - Length (9 bits): Field specifying the length of the TLV value field in octets (bytes), including the port description string D
 - Description: Variable-length field containing the human-readable port description

Organizationally Defined TLV Set

Organizationally Defined TLVs are custom TLVs defined by specific organizations to convey proprietary or specialized information not covered by standard TLVs. These TLVs enable the transmission of custom data between LLDP-enabled devices within an organization's network.

- Format
 - Type (7 bits): TLV type (typically 127-255)
 - Length (9 bits): Length of the TLV value field in octets OUI (3 octets): Organizationally Unique Identifier assigned by the IEEE
 - Subtype (1 octet): Specific purpose or format of the custom TLV
 - Custom Data (Variable Length): Custom data specific to the organization
- Component
 - Type: 7-bit TLV type indicator
 - Length: 9-bit length indicator
 - OUI: 3-octet Organizationally Unique Identifier
 - Subtype: 1-octet subtype indicator

LLDP Auto Sense FE Feature

Within the Organizationally Defined TLV framework, three new TLVs (subtypes 5, 6, and 7) have been introduced to facilitate FE connectivity and tunnel information, leveraging the unique identifier OUI D8:84:66.

LLDP TLV Type Extreme FE Connectivity (SubType 5)

This TLV is exclusively sent by the SD-WAN appliance and received by the Fabric Engine device. It contains the connectivity parameters. The connectivity parameters are:

- Switch VTEP IPv4 address (for VLAN 4047 and VTEP tunnel source IP)
- Gateway IPv4 address (for default route next-hop)
- MTU (default value: 1500)

The following parameters are not supported:

- Management interface
- Dot1q tag
- DNS
- BGP ASN of switch
- IPv4 BGP addresses

LLDP TLV Type Extreme FE VXLAN Tunnel (SubType 6)

This TLV is sent exclusively by the SDWAN device and received by the Fabric Engine device. It contains VTEP tunnel information, grouped by different ISIS metrics. Maximum of 8 different metrics and 3 TLVs (to accommodate up to 255 tunnels).

Each TLV contains:

- Number of tunnel groups
- Metric for each group (default: 20000)

- Tunnel state (0 = disabled, 1 = enabled)
- IPv4 addresses for logical-interfaces destination-IPs

The Fabric Engine device will

- Create ISIS logical-interfaces for all tunnels
- Enable/disable ISIS on interfaces based on tunnel state changes
- Delete logical-interfaces for tunnels no longer present in LLDP updates

LLDP TLV Type Extreme FE Adjacency SubType (7)

This subtype is exclusively sent by Fabric Engine devices. When received, the SD-WAN appliance extracts and updates its ISIS adjacency state based on the information contained in the TLV.

HMAC SHA Digest 256 Encoding

All Front End (FE) Type-Length-Values (TLVs) are secured using HMAC SHA-256 encryption.

To ensure seamless decryption and authentication, Fabric Engine and SDWAN appliances must share the same password. If a message is tampered, authentication will fail, and the TLV will be discarded without notification. Both Fabric Engine and SDWAN appliances can authenticate and drop TLVs with mismatched digests.

LLDP Neighbor Table Supported Objects

=====

```
/openconfig-lldp:lldp/interfaces/interface/neighbors
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/age
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/chassis-id
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/chassis-id-type
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/id
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/last-update
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/management-address
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/management-address-
type
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/port-description
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/port-id
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/port-id-type
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/system-description
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/system-name
/openconfig-lldp:lldp/interfaces/interface/neighbors/neighbor/state/ttl
```

LLDP Neighbor Table Unsupported Objects

=====

None

LLDP Stats Supported Objects

=====

/openconfig-lldp:lldp/state/counters/frame-in

/openconfig-lldp:lldp/state/counters/frame-out

/openconfig-lldp:lldp/state/counters/last-clear

/openconfig-lldp:lldp/interfaces/interface/state/counters/frame-in

/openconfig-lldp:lldp/interfaces/interface/state/counters/frame-out

/openconfig-lldp:lldp/interfaces/interface/state/counters/last-clear

/openconfig-lldp:lldp/interfaces/interface/state/counters/frame-discard

/openconfig-lldp:lldp/interfaces/interface/state/counters/frame-error-in

/openconfig-lldp:lldp/interfaces/interface/state/counters/frame-error-out

/openconfig-lldp:lldp/interfaces/interface/state/counters/tlv-discard

/openconfig-lldp:lldp/interfaces/interface/state/counters/tlv-unknown

/openconfig-lldp:lldp/state/counters/entries-aged-out

/openconfig-lldp:lldp/state/counters/frame-discard

/openconfig-lldp:lldp/state/counters/frame-error-in

/openconfig-lldp:lldp/state/counters/tlv-accepted

/openconfig-lldp:lldp/state/counters/tlv-discard

/openconfig-lldp:lldp/state/counters/tlv-unknown

LLDP Fabric Extend YANG Schema Paths

```
fabric-extend: /lldp/interfaces/interface[name=]/fabric-extend
  config: /lldp/interfaces/interface[name=]/fabric-extend/config
    advertise-fabric-ip: /lldp/interfaces/interface [name=]/fabric-extend/config/
advertise-fabric-ip
  authentication-password: /lldp/interfaces/interface[name=]/fabric-extend/
authentication-password
  enabled: /lldp/interfaces/interface[name=]/fabric-extend/config/enabled
  mtu: /lldp/interfaces/interface[name=]/fabric-extend/config/mtu
  prefix-length: /lldp/interfaces/interface[name=]/fabric-extend/config/prefix-length
  remote-neighbors: /lldp/interfaces/interface [name=]/fabric-extend/remote-neighbors
  remote-neighbor: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/
remote-neighbor [name=]
  config: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/config
  ip: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor [name=]/config/ip
```

```

isis-metric: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/
remote-neighbor[name=]/config/isis-metric
name: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/config/name
name: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/name
state: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/state
ip: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/config/ip
isis-metric: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/
remote-neighbor[name=]/config/isis-metric
name: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/state/name
status: /lldp/interfaces/interface[name=]/fabric-extend/remote-neighbors/remote-
neighbor[name=]/state/status
state: /lldp/interfaces/interface[name=]/fabric-extend/state
advertise-fabric-ip: /lldp/interfaces/interface[name=]/fabric-extend/state/
advertise-fabric-ip
authentication-password: /lldp/interfaces/interface[name=]/fabric-extend/state/
authentication-password
enabled: /lldp/interfaces/interface[name=]/fabric-extend/state/enabled
mtu: /lldp/interfaces/interface[name=]/fabric-extend/state/mtu
prefix-length: /lldp/interfaces/interface[name=]/fabric-extend/state/prefix-length

```

LLDP Custom TLV YANG Schema Path

```

custom-tlvs: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-tlvs
tlv: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-tlvs/
tlv[type=][oui=][oui-subtype=]
addresses: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/
custom-tlvs/tlv[type=][oui=][oui-subtype=]/addresses
address: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/
custom-tlvs/tlv[type=][oui=][oui-subtype=]/addresses/address[ip=]
ip: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/addresses/address[ip=]/ip
state: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/
custom-tlvs/tlv[type=][oui=][oui-subtype=]/addresses/address[ip=]/state
ip: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/addresses/address[ip=]/state/ip
isis-state: /lldp/interfaces/interface[name=]/neighbors/
neighbor[id=]/custom-tlvs/tlv[type=][oui=][oui-subtype=]/addresses/address[ip=]/state/
isis-state
config: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/config
oui: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-tlvs/
tlv[type=][oui=][oui-subtype=]/oui
oui-subtype: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/
custom-tlvs/tlv[type=][oui=][oui-subtype=]/oui-subtype
state: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/state
oui: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/state/oui
oui-subtype: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/
custom-tlvs/tlv[type=][oui=][oui-subtype=]/state/oui-subtype
type: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/state/type
value: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/state/value
type: /lldp/interfaces/interface[name=]/neighbors/neighbor[id=]/custom-
tlvs/tlv[type=][oui=][oui-subtype=]/type

```

Log Location

Service	Location
CLI	/var/log/tierra/cli/
APIGW	/var/log/tierra/api-gw/ /tmp/tierra/trace/api-gw/
LLDP	/tmp/tierra/trace/lldp
Docker	/var/log/containers/ /var/log/pods/

YANG Enhancements

The following fabric container is augmented under Openconfig-lldp/lldp/interfaces/ interface:

```

+--rw extr-lldp-ext: fabric-extend
  +--rw extr-lldp-ext:config
    | +--rw extr-lldp-ext:enabled?                               boolean
    | +--rw extr-lldp-ext: mtu?                                   uint16
    | +--rw extr-lldp-ext: advertise-fabric-ip?                 oc-inet: ip-address
    | +--rw extr-lldp-ext:prefix-length?                         uint8
    | +--rw extr-lldp-ext:authentication-password?              oc-types: routing-
password
  +--ro extr-lldp-ext: state
    | +--ro extr-lldp-ext:enabled?                               boolean
    | +--ro extr-lldp-ext:mtu?                                   uint16
    | +--ro extr-lldp-ext:advertise-fabric-ip?                  oc-inet: ip-address
    | +--ro extr-lldp-ext:prefix-length?                         uint8
    | +--ro extr-lldp-ext: authentication-password?              oc-types: routing-
password
  +--rw extr-lldp-ext:remote-neighbors
    +--rw extr-lldp-ext:remote-neighbor* [name]
      +--rw extr-lldp-ext:name                                   -> ../config/name
      +--rw extr-lldp-ext:config
        | +--rw extr-lldp-ext:name?                             string
        | +--rw extr-lldp-ext:ip?                               oc-inet: ip-address
        | +--rw extr-lldp-ext:isis-metric?                      uint32
      +--ro extr-lldp-ext: state
        +--ro extr-lldp-ext:name?                               string
        +--ro extr-lldp-ext:ip?                                 oc-inet: ip-address
        +--ro extr-lldp-ext:isis-metric?                        uint32
        +--ro extr-lldp-ext:status?                             boolean

```

LLDP Configuration Guidelines and Restrictions

- The Extreme Networks implementation of LLDP supports standard LLDP information.
- Mandatory TLVs are always advertised.
- The exchange of LLDP link-level parameters is transparent to the other Layer 2 protocols. The LLDP link-level parameters are reported by LLDP to other interested protocols.

Configuring and Managing LLDP

The following sections discuss working with the Link Layer Discovery Protocol (LLDP) on Extreme Networks devices.

Understanding the Default LLDP

The following table lists the default LLDP configuration. Consider this when making changes to the defaults.

Table 7: Default LLDP Configuration

Parameter	Default setting
LLDP global state	Enabled
LLDP receive	Enabled
LLDP transmit	Enabled
Transmission frequency of LLDP updates	30 seconds
Hold time for receiving devices before discarding	120 seconds

Disabling LLDP Globally

To globally disable LLDP, perform the following steps:

1. From privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Disable LLDP globally.

```
device(config)# no protocol lldp
```

3. Verify LLDP global configuration.

```
device(config-lldp)# show running-config lldp
no protocol lldp
system-description Extreme BR-Extreme 8730 Router
disable
!
```

The following configuration is an example of the previous steps to disable LLDP.

```
device# configure terminal
device(config)# no protocol lldp
```

If required, re-enable LLDP.

```
device(config)# protocol lldp
```

Configuring LLDP Global Parameters

The following LLDP global parameters are available:

```
device(config)# lldp
device(config-lldp)#
end                End current mode and change to enable mode
exit              Exit current mode and down to parent mode
hello-timer       LLDP hello-timer configuration in seconds
```

holdtime	LLDP hold time configuration in seconds
list	List all supported configuration commands
management-address	LLDP management IP address configuration
no	Negate a command or set its defaults
pwd	Display current mode
receive	Enable LLDP receive mode for all the ports
suppress-tlv	Select lldp suppress TLVs
transmit	Enable LLDP transmit mode for all the ports

Enabling and Disabling the Receiving and Transmitting of LLDP Frames

By default both transmit and receive for LLDP frames is enabled.

- The following example enables only receiving of LLDP frames.

```
device(config-lldp)# receive
```

- The following example enables only transmitting of LLDP frames.

```
device(config-lldp)# transmit
```

Configuring the Transmit Frequency of LLDP Frames

The default transmit frequency of LLDP frames is 30 seconds. You can change the frequency from 1 to 65535 seconds. The following example changes the frequency to 45 seconds.

```
device(config-lldp)# hello-timer 45
```

Configuring the Hold Time for Receiving Devices

The default hold time for storing peer information from LLDP packets is 30 seconds. The example below demonstrates how to modify this hold time to 60 seconds:

```
device(config-lldp)# holdtime 60
```

Advertising the Optional LLDP TLVs

By default, LLDP advertises the optional TLVs for management address, port description, system description, and system name. However, you can choose to suppress these TLVs. The following example illustrates how to suppress the management address and system description TLVs:

```
device(config-lldp)# suppress-tlv management-address system-description
```

Displaying LLDP Information

The **show lldp** command allows you to display the following information:

- LLDP status
- LLDP neighbor information
- LLDP statistics

Displaying LLDP Status

To display the global LLDP status, use the **show lldp** command.

```
device# show lldp
LLDP Config Info

Global LLDP Information:
```

```
Status: INACTIVE
LLDP advertisements are sent every 200 seconds
LLDP hold time is 120 seconds
LLDP transmit: Off
LLDP receive: On
LLDP Static Mgmtip: 1.1.1.1
LLDP Suppress Tlvs:
```

Interface LLDP Information:

Intf	State	Rx	Tx
management 0	Enabled	On	On
ethernet 0/1	Disabled	On	On
ethernet 0/2	Disabled	On	On
ethernet 0/3	Disabled	On	On
ethernet 0/4	Disabled	On	On
ethernet 0/5	Disabled	On	On
ethernet 0/6	Disabled	On	On
ethernet 0/7	Disabled	On	On
ethernet 0/8	Disabled	On	On
ethernet 0/9	Disabled	On	On
ethernet 0/10	Disabled	On	On
ethernet 0/11	Disabled	On	On
ethernet 0/12	Disabled	On	On
ethernet 0/13	Enabled	On	On

To display LLDP status for an Ethernet interface, use the **show lldp interface ethernet** command.

```
device# show lldp interface ethernet 0/11:1
LLDP Interface Information:
-----

Interface: ethernet 0/11:1
Status: Enabled
Transmit: On
Receive: On
Static Mgmtip: Disabled
Suppressed Tlvs:
device#
```

Displaying LLDP Neighbor Information

To display the LLDP neighbor information, use the **show lldp neighbors** command. This command allows you to display the information for all Ethernet interfaces, a specific interface, or detailed neighbor information.

The following example displays the LLDP neighbor information for all interfaces.

```
device# show lldp neighbors
sh ll neighbors
show LLDP Neighbors Information
```

Chassis ID	Local-Port	Dead Intvl	Rem Life	Remote Port-ID	Remote Port Descr	System Name
Switch 2426	1/18	180	100	Ethernet0/26	LLDP-1	DUT-1
Switch 2426	1/22	120	50	Ethernet0/27	LLDP-2	DUT-2

```
Total entries displayed: 2
```


The following example displays the LLDP neighbor information for Ethernet interface 0/11:1.

```
device# show lldp neighbors interface ethernet 0/11:1
LLDP Interface Neighbors:
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Interface: ethernet 0/11:1
Chassis id: 40:88:2f:fa:fc:00
Remote Port id: ethernet 0/29:1
Remote Port Description: Ethernet 0/29:1
System Name: 8730-1
System Description: Extreme 8720-32C, 5.10.210-yocto-standard, Version Ex \
                    tremeOneBase-10.1.2.1-008
Dead Interval: 120 seconds
Time remaining: 113 seconds
System Capabilities:
Enabled Capabilities:
Management Address: 10.38.63.162
Management Address IPV6: not advertised
device#
```

Displaying LLDP Statistics

To display the LLDP statistics for all interfaces or a specific interface, use the **show counters lldp** command. The following example displays the statistics for Ethernet interface 0/18.

```
device#show counters lldp summary
LLDP Global Statistics:
    FrameIn: 110299
    FrameOut: 122258
EntriesAgedOut: 0
FrameDiscard: 0
FrameErrorIn: 0
    TlvAccepted: 110299
    TlvDiscard: 0
    TlvUnknown: 16694
    LastClear: 0s

device# show counters lldp interface ethernet 0/1:1
LLDP Interface Statistics: ethernet 0/1:1
    FrameIn:
    FrameOut: 2250
    FrameDiscard:
    FrameErrorIn:
    FrameErrorOut:
    TlvDiscard:
    TlvUnknown:
    LastClear: 0s
```

Use the **show counters lldp interface ethernet all** command to display the statistics for all interfaces.

Clearing LLDP-Related Information

To clear LLDP-related information, perform the following steps.

1. Clear the LLDP statistics on an interface.

```
device# clear counters lldp interface ethernet 0/11:1
```

This example clears the LLDP transmit and receive counters on the Ethernet interface 1/8.

2. Clear the LLDP statistics for all interfaces.

```
device# clear counters lldp all
```



gNMI and gNOI

[gNMI and gNOI Overview](#) on page 99
[gRPC Network Management Interface \(gNMI\) Service](#) on page 101
[gRPC Network Operational Interface \(gNOI\) Service](#) on page 104
[Configure gRPC Server](#) on page 104
[Data Model and Northbound Interface](#) on page 105
[Storage of List Key Values](#) on page 106
[Inband Management Support](#) on page 106
[CLI Commands](#) on page 106
[Event Log Messages](#) on page 107
[gNMI Authentication and Encryption](#) on page 107
[Importing gNMI Client CA Root Certificate](#) on page 107
[Security](#) on page 108

gNMI and gNOI Overview

Extreme ONE OS platform supports gNMI, gNOI and gNSI (gRPC services).



Note

- Multiple gRPC server instances can be configured
- Each gRPC server instance can be configured to run with a unique combination of port, certificate and VRF
- Support the import of external certificates and associate them with the gRPC server instance..
- gNMI does not support Poll subscriptions.
- ONIE-install removes all certificates from the device. If you restore a previously backed-up configuration (like a gRPC server setup that uses a certificate), it will fail due to missing certificates. To resolve this, you must re-import or regenerate the required certificates before applying the configuration

Troubleshooting Information

- To enable gRPC services,
 - Ensure at least one gRPC server instance is configured and operational.

- Verify the gRPC server's status using the command `show running-state system grpc-server`.
- For issues with gNMI/gNOI clients,
 - Use open-source clients like `gnmic` with the `--debug` option for connection insights.
 - For proprietary clients, check error codes for issue understanding.

**Note**

Client-side context determines the gRPC client connection timeout if a gNMI client subscription doesn't consume the response for more than 10 seconds, then the gRPC subscription is cancelled.

- For device-level issues,
 - Check Service-level RasTrace logs in `show trace <service name>`.
 - Collect API Gateway, Security, and Ingress Gateway logs from this location.
 - View audit logs for configuration commands using `show logging audit config`.

gNMI Overview

What is gNMI?

gNMI (gRPC Network Management Interface) is a modern network management protocol developed by the OpenConfig working group. It is designed to standardize the way network devices are configured and monitored, using efficient, secure, and scalable mechanisms. gNMI uses HTTP/2 as its transport protocol.

Key Features of gNMI

1. **Built on gRPC:** gNMI is built on top of gRPC (Google Remote Procedure Call), a high-performance, open-source RPC framework. gRPC uses a compact and efficient binary protocol (Protocol Buffers), which is significantly more efficient than text-based protocols like REST (which commonly use JSON or XML). This results in reduced bandwidth usage and faster data exchange between network devices and management systems.
2. **Data Modeling with YANG:** gNMI uses OpenConfig YANG models to define the structure of configuration and operational data. YANG is a standardized data modeling language that ensures consistency and interoperability across different vendors and platforms.
3. **Real-Time Streaming Telemetry:** gNMI supports streaming telemetry, allowing devices to continuously push real-time updates to subscribed clients. This is ideal for proactive monitoring, performance analysis, and rapid fault detection, moving away from inefficient polling methods.
4. **Enhanced Security:** gNMI secures communication using TLS (Transport Layer Security) for encryption and authentication. It also supports mutual TLS (mTLS), which ensures both the client and server authenticate each other using digital certificates, providing strong end-to-end security.
5. **Purpose-Built for Network Management:** Unlike traditional protocols like SNMP or generic interfaces like REST, gNMI is designed specifically for network configuration

and state management. It supports robust operations such as Get, Set, Subscribe, and Capabilities, offering a scalable and vendor-agnostic interface.

gNOI Overview

What is gNOI?

gNOI (gRPC Network Operations Interface) is a modern, gRPC-based protocol developed by the OpenConfig working group. It is designed specifically for performing operational and lifecycle management tasks on network devices, complementing configuration and telemetry protocols like gNMI.

Key Features of gNOI

1. gRPC-Based and High Performance: gNOI is built on the gRPC framework, enabling efficient, bi-directional communication using HTTP/2 and Protocol Buffers (Protobuf).

This architecture ensures low latency, high throughput, and efficient binary serialization, making it ideal for real-time operational workflows.

2. Operational and Lifecycle Focus: Unlike gNMI, which handles configuration and telemetry, gNOI is focused solely on operational tasks such as certificate management, software installation, factory reset, boot control, and system time synchronization.

It provides well-defined RPCs to perform these tasks in a programmatic and vendor-agnostic way.

3. Extensible Architecture: gNOI is designed to be extensible, allowing new operational RPCs to be added as network management requirements evolve. This makes it future-proof and adaptable to modern operational needs.
4. Standardized and Vendor-Neutral: gNOI ensures a consistent operational interface across devices from different vendors, simplifying automation, orchestration, and tooling.

It helps eliminate the need for vendor-specific CLIs or proprietary APIs for routine operational tasks.

gRPC Network Management Interface (gNMI) Service

gNMI (gRPC Network Management Interface) is a management protocol that enables efficient configuration management and telemetry data collection from network devices. Key features include: - Modification and retrieval of device configurations - Control and generation of telemetry streams - Compatibility with OpenConfig YANG data models.

gNMI offers a scalable, secure, and real-time monitoring solution, employing a push-based model for data streaming. This approach differs from traditional pull-based protocols like SNMP. For gNMI specification, see [reference/rpc/gNMI](#).

Capabilities RPC

Extreme ONE OS support gNMI version 0.10.0.

A client can use the Capabilities RPC to discover the target's capabilities. The target provides details such as the gNMI service version, supported YANG data modules, and available data encodings. This information helps the client specify the set of models (for Get, Set, or Subscribe RPCs) and the data encoding to be used in subsequent RPC messages.

At Extreme ONE OS rendering of capabilities for YANG data modules also includes a list of module names that are internally imported by other YANG modules. The capability information can also be retrieved from CLI using exec level command ``show grpc-server gnmi capabilities``.

Get RPC

The Get RPC allows a client to request a snapshot of configuration or state data by specifying a set of paths from the data tree.



Note

- The Get RPC is designed for retrieving relatively small data sets. It is not ideal for large-scale data retrieval, such as the entire component inventory. For such cases, use Subscribe RPC..
- Wildcard-based Get operations are not supported. For example, a Get request on a path like `/interfaces/interface[name=*]` is not allowed

Set RPC

The Set RPC allows clients to modify the target's state by sending a request with desired changes. Supported operations include delete, update, and replace. The server processes these operations in the following order: delete, replace, and then update, when all operations are grouped together in a single Set request.

Set RPC allows bulking of configurations, upon receiving the request, API Gateway leverages Openconfig data model definitions to perform syntactic and semantic validations. If validation fails, entire configuration is rejected, and an appropriate error is returned to the client. Upon successful validation, the configuration data is stored in the database.

Key Points:

- Bulk configuration, including single or multiple modules, is supported through a single gNMI Set RPC. Bulk configurations are limited by the default gRPC size constraint of approximately 4MB.
- Wildcard-based Set operations are not supported. For instance, updating the MTU attribute for all interfaces is not supported.

- Any failure during the data validation phase (including syntactic and semantic errors) will result in the entire Set request failing, and an appropriate error will be returned to the client.

Delete Configuration

Upon receiving a delete request, the server removes the specified path from the target's data tree. If the path includes child elements, they must be recursively deleted. If the specified path does not exist, the delete operation is silently accepted.

Replace Configuration

- If the client specifies a path-value, the server replaces it with the new value.
- If no value is provided and there is no default in the schema, the path must be deleted.

Update Configuration

When a client sends a Set RPC request, the target device will:

1. Create a new data tree element if the specified path doesn't exist, as long as the path is schema-compliant
2. Populate the element with the provided data.

Subscribe RPC

Clients can receive real-time updates about target data through the Subscribe RPC. A subscription requires:

1. Specifying one or more data paths
2. Choosing a subscription mode (periodic or event-driven)

This approach, known as streaming telemetry, replaces traditional polling with a more efficient push model. By leveraging a standardized data model over gRPC transport, clients can enjoy additional benefits, including, Real-time updates, Reduced latency, and Improved efficiency

Once Subscriptions

In ONCE mode, the subscription operates as a single request/response. The target sends the relevant updates and then closes the RPC.

Stream Subscriptions

Stream subscriptions provide ongoing updates for selected paths until cancelled, ensuring continuous access to new information. Stream mode supports the following options:

- **On Change (ON_CHANGE):** Updates are sent only when the value of the data item changes. Initially, the target sends updates for all subscribed paths, and afterward, only when values change. A heartbeat interval can be specified, in which case, the data will be re-sent at the specified interval, even if the value hasn't changed.
- **Sampled (SAMPLE):** Updates are sent at regular intervals, defined by the sample interval. If the sample interval is set to 0, the target sends updates at the shortest possible interval.

Key Points

- Wildcard-based Subscribe operations are supported. For example, a Subscribe request on a path like `/interfaces/interface[name=*]` is allowed.
- Certain restrictions will be imposed on the minimum interval that can be set for interval-based stream subscriptions.
- A few YANG paths will not be exposed for on-change subscriptions to reduce the overall system load.

gRPC Network Operational Interface (gNOI) Service

gNOI (gRPC Network Operations Interface) is a protocol within the gRPC ecosystem designed to facilitate operational and administrative tasks on network devices. It complements gNMI (gRPC Network Management Interface), which primarily focuses on configuration and state management, by addressing the operational needs of network management. Supported gNOI version is v0.4.0.

gNOI vs gNMI

Aspect	gNOI	gNMI
Purpose	Operational and lifecycle management	Configuration and state management
Tasks	Software upgrades, certificate management, diagnostics	Configuration, telemetry, and subscription management
Scope	Focused on actions	Focused on data

Configure gRPC Server

The gRPC server is disabled by default. Follow the procedure to configure a gRPC server.



Note

After enabling gRPC server, changes to attributes require a service restart: disable with 'no enable' and then re-enable for the changes to take effect. Multiple instances can be configured with unique port, VRF, and certificate combinations.

1. Generate grpc app certificate either by generating the certificate from the device or by importing a custom certificate, and then enable the gRPC server.

Generate grpc app certificate:

```
certificate-manager generate ssl-profile-id ssl-reserved-generated certificate-
extension san 1.1.1.1 (1.1.1.1 is device ip)
```

Or import any custom certificate and associate it with the ssl-profile-id ssl-reserved-generated.

2. Run the **grpc-server <NAME>** command to configure a gRPC server.

This command creates a gRPC server instance, defaulting to "DEFAULT" if no name is specified. Configure the port and certificate ID for use. If no VRF is specified, the system will default to mgmt-vrf.

```
device# configure terminal
device(config)# system
device(config-system)# grpc-server
device(config-system-grpc-server-DEFAULT)#
device(config-system-grpc-server-DEFAULT)# port 443
device(config-system-grpc-server-DEFAULT)# certificate-id ssl-reserved-generated
device(config-system-grpc-server-DEFAULT)# enable
device(config-system-grpc-server-DEFAULT)#
```

The gRPC server is up and running. Users can use clients like gnmic for gNMI set, subscribe, and on-change operations.

3. Configure mutual TLS.

This command enables mutual exchange of x509 certificate-based authentication for the gRPC client. To update the gRPC server configuration parameters, disable the gRPC server first, make the changes, and then re-enable it.

```
device# configure terminal
device(config)# system
device(config-system)# grpc-server mygrpcserver
device(config-system-grpc-server-mygrpcserver)# certificate-id test_cert_id
device(config-system-grpc-server-mygrpcserver)# port 9340
device(config-system-grpc-server-DEFAULT)# no enable
device(config-system-grpc-server-mygrpcserver)# mutual-tls
device(config-system-grpc-server-mygrpcserver)# enable
device(config-system-grpc-server-mygrpcserver)#
```

4. Create a gRPC instance with a custom VRF.

```
device(config)# system
device(config-system)# grpc-server
device(config-system-grpc-server-DEFAULT)# vrf vrfnew1
%Error: gRPC server DEFAULT is in enabled state, Please disable for altering any
configuration
device(config-system-grpc-server-DEFAULT)# no enable
device(config-system-grpc-server-DEFAULT)# vrf vrfnew1
device(config-system-grpc-server-DEFAULT)# enable
```

5. Verify the configured gRPC server.

```
device# show running-config system grpc-server
system
  grpc-server mygrpcserver
    certificate-id test_cert_id
    port 9340
    mutual-tls
    vrf vrfnew1
    enable
  !
|
device#
```

Data Model and Northbound Interface

Extreme ONE OS and its supported applications rely on OpenConfig YANG modules as their core data model. These modules adhere to a strict style guide, which clearly separates configurable and operational data and organizes attributes into distinct hierarchical structures. This style guide ensures consistency in YANG module definitions, promoting clarity and simplicity.

Key aspects of the data model:

- Based on OpenConfig YANG modules
- Distinguishes between "config" and "state" data
- Organizes data hierarchically

Storage of List Key Values

YANG key values are stored in a case-sensitive manner, treating uppercase and lowercase characters as distinct. However, exceptions apply to MAC addresses and IPv6 address configurations, which are stored in lowercase.

Inband Management Support

The system allows users to configure multiple gRPC servers, each operating within its own unique Virtual Routing and Forwarding (VRF) context. Additionally, external clients can access the device for configuration and retrieval operations via either the default VRF or user-defined VRFs.

CLI Commands

For more information about syntax and parameters, see *Extreme ONE OS Switching Command Reference Guide*.

Config Commands

- `grpc-server`
- `grpc-server - certificate-id`
- `grpc-server - port`
- `grpc-server - vrf`
- `grpc-server - enable`
- `grpc-server - mutual-tls`

Exec Commands

- `show grpc-server gnmi statistics`
- `show grpc-server gnmi capabilities`

Event Log Messages

gRPC server utilizes a shared logging and event framework to record messages and generate standardized events, providing insights into progress, status updates, errors, and failures.

Event ID	Event Log	Remarks
6001	gRPC request executed successfully	Indicates that the gRPC request has been executed successfully.
6007	Requisite flags for apigw service not set	Requisite flags for API Gateway not set, cannot proceed with service instantiation.
6009	Failed initializing microservice	Indicates API Gateway bring up failure
12002	gRPC server connection status	Indicates gRPC server instance is ready to process RPC requests
12003	gRPC server termination status	Indicates termination of gRPC server instance

gNMI Authentication and Encryption

Network devices utilizing the gRPC Network Management Interface (gNMI) must ensure secure, bidirectional communication over a gRPC channel while implementing standard authorization and accounting for all management operations. The figure below illustrates key components of a gNMI-based management system. In this setup, the configuration system and telemetry collectors function as RPC clients to the network device (target). The target, in turn, provides the gNMI service, offering methods for subscribing to telemetry streams and sending configuration data.

Credentials and Authentication

When configuring credentials and authentication, all operations are transmitted over an encrypted connection. Each gRPC message contains username and password credentials in its metadata. To make configuration changes, you need a user with Read-Write permissions.

The target device uses these credentials to authorize configuration operations, employing available AAA (Authentication, Authorization, and Accounting) methods. The network element's local AAA configuration determines how and where AAA requests are sent.

For example, if a network element is set up to use TACACS for command authorization, a `Get(/interfaces)` request with a username and password will trigger a TACACS command authorization request using the same credentials.

Importing gNMI Client CA Root Certificate

It is assumed that your infrastructure is set up with a gNMI client. Its configuration is beyond the scope of this document.

Importing the gNMI client's CA Root Certificate for password less connection between the gNMI Server and Client.

The CA certificate of the gNMI client is imported from the remote server. Use the **certificate-manager** command to import, generate, export, and delete the certificates.

```
# certificate-manager ?  
delete      Delete certificate  
export      Export Certificate to remote server  
generate     Generate switch Certificate for server application  
import       Import Certificate and Key  
import-pkcs  Import pkcs certificate key bundle
```

Type ? after executing a command to display help for the command.

Security

The gRPC API Gateway ensures secure, bidirectional communication over a gRPC channel, following the "gNMI Authentication and Encryption" specifications.

TLS encryption is used for sessions between the gNMI server and client, with mutual authentication for new connections. Each entity verifies the remote entity's X.509 certificate for recognition and authorization.

Encryption

- TLS 1.3 is used by default, with a minimum supported version of TLS 1.2.
- The gRPC channel must be encrypted before use
- Unencrypted communication is prohibited; channel creation will fail if encryption cannot be established.

Authentication

- Certificate-based validation confirms the connecting network management system is an authorized endpoint
- Both user and token-based authentication are supported.



BMC Configuration

[Increase BMC Security](#) on page 109

[Change BMC User Password](#) on page 110

[Enable the BMC Management Interface](#) on page 110

[Configure BMC Management Interface IP Address](#) on page 111

[Reset BMC Configuration to Factory Defaults](#) on page 113

Increase BMC Security

This topic discusses the steps to increase BMC's security including changing the password for the default account, and changing the default IP address for the BMC's network interface.

Intelligent Platform Management Interface

Intelligent Platform Management Interface (IPMI) is a set of specifications that defines how to manage and monitor a device independent of its Operating System (OS), underlying Hardware, and the BIOS installed on it. IPMI also defines a set of physical interfaces that enable system administrators to perform *out-of-band* management of IPMI capable devices, including such devices that have been powered off or that have network issues or are unresponsive. Without IPMI, a system administrator would need to be physically present near the device to resolve any issue.

IPMI is a message-based, hardware-level interface specification which exists and operates independently of the underlying operating system or the device's hardware. This enables IPMI to remotely manage a device even if the device does not have an installed OS. IPMI can also be used in scenarios where the device is powered down or even if there is a system or OS failure.

The target device can be powered down, however, for IPMI to work, it must at least be connected to an underlying local area network (LAN) and must be connected to a working power source.

IPMI can also be used to continuously monitor various statuses and statistics, such as temperature, fan speed, voltages, power supply status and physical access to the device.

Baseboard Management Controller

Baseboard Management Controller (BMC) is a dedicated microcontroller embedded on a device's motherboard and has its own dedicated firmware, RAM, and network port. Sensors on the motherboard transmit data to the BMC which in turn transmits this data to dedicated centralized monitoring servers. When connected to a LAN, the network port on the BMC enables *out-of-band* control and monitoring of the underlying hardware.

BMC enables IPMI on a device.

Securing BMC

BMC ships with a well known default User ID, password, and network configuration configured during firmware install at the factory. This provides an security vulnerability that can be exploited to gain access to the device.

Securing BMC involves changing the default User ID's password and changing the default network configuration. ONE OS provides commands that interacts with the underline BMC firmware to harden the security of your device's BMC.

Change BMC User Password

Configuration of each User ID must be done separately from within its configuration mode.



Note

Only the user with the User ID 2 can be configured. Though the command allows configuration of other User IDs, those changes are not applied on the BMC.

To change the password for a specific BMC user:

To change the password for user ID 2 on the BMC of the device, run the following command:

```
device# bmc user 2 password 1#bmcPasswd0!  
Set User Password command successful (user 2)  
device#
```

Enable the BMC Management Interface

Follow the procedure to enable the BMC Management Interface.

1. Navigate to the BMC Management Interface context.

```
device # configure terminal  
device(config)#  
device(config)# bmc  
device(bmc)#  
end          End current mode and change to enable mode  
exit         Exit current mode and down to previous mode  
interface    BMC management interface
```

list	Print command list
pwd	Display current mode path

**Note**

Only the BMC Management Interface with interface ID of 0 (zero) can be configured.

You are now within the BMC configuration mode.

- From within the BMC configuration mode, navigate to the BMC Management Interface configuration mode.

```
device(config)# bmc
device(bmc)# interface 0
device(interface-0)#
end          End current mode and change to enable mode
exit         Exit current mode and down to previous mode
ipv4        The internet protocol (ipv4) information
list        Print command list
no          Negate a command or set its defaults
pwd         Display current mode path
shutdown    Shut down the BMC management interface
```

- Enable the interface.

```
device(interface-0)#no shutdown
```

The BMC Management Interface is now enabled and ready to be configured for out-of-band access.

- (Optional) Verify if the BMC Management Interface is enabled.

```
device# show running-config bmc
!
bmc
  interface 0
    ip-address 192.168.1.1
    ip-gateway 192.168.1.2
    ip-mode dhcp netmask 255.255.254.0
    enable false
!
!
device#
```

Configure BMC Management Interface IP Address

Keep the IPv4 address, the netmask, and the Gateway IPv4 address that is required to be configured, ready.

**Note**

Extreme ONE OS configures the following static IP 192.x.x.x/24 and default gateway 0.0.0.0 by default.

To configure the BMC Management Interface:

1. Navigate into the BMC Management Interface context.

```
device# configure terminal
device(config)# bmc
device(bmc)#
```



Note

Only the BMC Management Interface with interface ID of 0 can be configured.

You are now within the BMC configuration mode.

2. From within the BMC configuration mode, navigate into the BMC Management Interface configuration mode.

```
device(bmc)# interface 0
device(interface-0)#
```

You are now within the BMC Management Interface configuration mode.

3. Configure the IPv4 address for the BMC Management Interface.

- To configure the BMC Management Interface to receive the IPv4 address from a remote DHCP server, run the following command:

```
device(config)# bmc
device(bmc)# interface 0
device(interface-0)# ipv4 address dhcp
```

- To configure the BMC Management Interface's IPv4 address manually, run the following command:

```
device(config)# bmc
device(bmc)# interface 0
device(interface-0)# ipv4 address 192.x.x.x/xx
device(interface-0)# ipv4 gateway 192.x.x.x
```

The IPv4 address for the BMC Management Interface is either automatically assigned or manually configured.

4. (Optional) Verify by issuing the **show running-config bmc** command.

```
device(bmc)# show running-config bmc
!
bmc
  device(interface-0)#
device(interface-0)# ipv4 address 192.x.x.x/24
device(interface-0)# ipv4 gateway 192.x.x.x
device(interface-0)# no shutdown
device(interface-0)#
device # show running-config bmc
!
bmc
  interface 0
    no ipv4 address dhcp
    ipv4 address 192.x.x.x/24
    ipv4 gateway 192.x.x.x
    no shutdown
  !
!
device #
```


Reset BMC Configuration to Factory Defaults

To reset the device BMC configuration back to its factory defaults:

From the Executable Mode, execute the `bmc factory reset` command.

```
# bmc factory-reset
```

After receiving the factory reset command, a prompt will appear asking if you want to proceed. If you confirm (Y), the device will reboot and the BMC configuration will be restored to its factory default settings.

The following example shows the reset to BMC factory default settings.

```
# bmc factory reset
Extreme ONE OS needs to be restarted for normal operations, after bmc factory reset.
Do you want to continue? (y/n): y
```



Port Mirroring

[Mirroring Overview](#) on page 114

[Configuration Validations for Mirroring](#) on page 115

[YANG Modules for Mirroring](#) on page 115

[CLI Commands for Mirroring](#) on page 117

[Use Case Scenarios and Configuration Examples for Port Mirroring](#) on page 117

Use this topic to learn about the configuration and implementation of port mirroring in Extreme ONE OS.

Mirroring Overview

Mirroring duplicates traffic from a source interface to a mirror session, which is associated with a destination interface or interfaces. This feature is supported in both ingress and egress directions, allowing for the duplication of traffic on the source and sending it to the configured destination. The key components include VxLAN tunnels, Bridge domain, and Host mapping.

Types of Mirroring Supported

Local SPAN: Mirrors traffic from one or more interfaces on the switch to one or more interfaces on the same switch.

Port mirroring is supported only on Extreme 8730-32D hardware platforms.

Limitations

- Mirror Destination Interfaces: Only physical and internal interfaces are supported as mirror destination interfaces.
- Egress Mirroring: Egress mirroring happens in the ingress pipeline after IFP stage and may not capture CPU-injected packets.

Configuration Guidelines

- SPAN Destination Port: Should not be configured to carry normal traffic.
- Speed Mismatch: When mirroring with different speeds on span source and destination interfaces, traffic exceeding the destination interface's capacity will be dropped.

- Port-Based Mirroring: Only physical, internal, and port-channel interfaces are supported as sources.
- A maximum of 4 mirror sessions are supported.

Configuration Validations for Mirroring

The system enforces the following configuration validations for mirroring:

Single Destination Interface per Mirror Session

Each mirror session supports only one destination interface. Configuring more than one destination interface will trigger an error. The destination can be an Ethernet interface, a port-channel, or an internal interface.

```
48xt(config)# mirror mirror_session_1
48xt(config-mirror)# destination interface ethernet 0/3
48xt(config-mirror)# destination interface ethernet 0/4
Error: Mirror interface is already configured
```

No Port-Channel Members as Destination Interfaces

Interfaces that are part of a port-channel cannot be used as mirroring destination interfaces.

```
48xt(config)# mirror mirror_session_2
48xt(config-mirror)# destination interface ethernet 0/49:4
Error: Port-channel configuration is present on the interface.
```

No Port-Channel Members as Source Interfaces

Interfaces that are part of a port-channel cannot be used as mirroring source interfaces.

```
48xt(config)# int ethernet 0/49:4
48xt(config-if-eth-0/49:4)# mirror m1 in
Error: Port-channel configuration is present on the interface
```

You can configure the port-channel interface as a source for traffic mirroring.

```
32d(config-mirror)# exit
32d(config)# interface port-channel 1
32d(config-if-po-1)# mirror mirror_1 in
32d(config-if-po-1)# mirror mirror_1 out
```

YANG Modules for Mirroring

YANG modules provide a standardized way to configure and manage mirroring functionality on the system. There are multiple YANG modules involved:

Extreme Mirror YANG Module

This module defines the structure for mirroring configuration, including:

- Mirror name and description
- Truncate settings

- Destination interfaces and encapsulation settings (for example, VXLAN, GRE, ERSPAN)

```

module: extreme-mirror
  +--rw mirrors
    +--rw mirror* [name]
      +--rw name                                -> ../config/name
      +--rw config
        | +--rw name?                          string
        | +--rw description?                   string
        +--ro state
          | +--ro name?                        string
          | +--ro mirror-id?                   uint32
        +--rw destination
          | +--rw interfaces
          |   +--rw interface* [name]           /* Physical/LAG/Tunnel ... */
          |     +--rw interface                -> ../config/name
          |     +--rw config
          |       | +--rw name?                 string
          |       | +--ro state
          |       | +--ro name?                 string

```

OpenConfig ACL YANG Augmentation

This module augments the OpenConfig ACL YANG module to support mirroring. It adds a mirror action to ACL entries.

```

module: openconfig-acl
  +--rw acl
    +--rw acl-sets
      +--rw acl-set* [name type]
        +--rw name                            -> ../config/name
        +--rw type                            -> ../config/type
      +--rw acl-entries
        +--rw acl-entry* [sequence-id]
          +--rw sequence-id                    -> ../config/sequence-id
          +--rw actions
            +--rw config
              | +--rw forwarding-action        identityref
              | +--rw log-action?              identityref
              | +--rw extr-acl-ext:count?      boolean
              | +--rw extr-acl-ext:mirror?     string
            +--ro state
              +--ro forwarding-action          identityref
              +--ro log-action?                identityref
              +--ro extr-acl-ext:count?        boolean
              +--ro extr-acl-ext:mirror?       string

```

OpenConfig Interfaces YANG Augmentation

This module augments the OpenConfig Interfaces YANG module to support mirroring on interfaces. It defines the structure for mirror configuration on interfaces, including the mirror name and direction (ingress or egress).

```

augment /oc-if:interfaces/oc-if:interface:
  +--rw mirrors
    +--rw mirror* [name direction]
      +--rw name                          string
      +--rw direction                    identityref
      +--rw config
        | +--rw name?                    string

```

```
|  +--rw direction?  identityref
+--rw state
  +--rw name?       string
  +--rw direction?  identityref
```

CLI Commands for Mirroring

The system provides various CLI commands to configure mirroring functionality.

Mirror Destination Creation

1. Create a mirror session.

The `mirror <NAME>` creates a new mirror session.

```
device(config)# mirror <NAME>
```

2. Create a mirror description.

```
device(config)# mirror mirror_1
device(config-mirror)# description test mirror
device(config-mirror)#
```

3. Specify the destination interface.

The command sets the destination interface for the mirror session.

```
device(config-mirror)# destination interface (ethernet | internal ) <NAME>
```

Port-Based Mirroring

To configure port-based mirroring, use the following CLI command:

1. Enter interface configuration mode for the specified interface.

```
device(config)# interface ethernet 0/1
```

2. Configure mirroring.

The command configures mirroring on the interface, specifying the direction (ingress or egress) for the mirror session.

```
device(config-interface) mirror < MIRROR_SESSION_NAME > (in | out)
```

Use Case Scenarios and Configuration Examples for Port Mirroring

The following use cases demonstrate the example configuration of mirroring sessions on a network device.

Local SPAN Port-Based Mirroring

Physical, internal, and port channel interfaces can be configured as source or destination interfaces.

1. Create a mirror session with a physical interface as the mirror destination.

```
device(config)# mirror mirror_session_5
device(config-mirror)# destination interface ethernet 0/6
device(config-mirror)#
```

2. Create a mirror session with a port channel interface as the mirror destination.

```
device(config)# mirror mirror_session_6
device(config-mirror)# destination interface port-channel 20
device(config-mirror)#
```

3. Attach the mirror session to a physical interface.

```
device(config)# interface ethernet 0/8
device(config-if-eth-0/8)# mirror mirror_session_6 in
device(config-if-eth-0/8)# mirror mirror_session_6 out
device(config-if-eth-0/8)#
```

4. Display the destination interface where mirror is created.

```
device# show running-config mirror
mirror m1
set interface ethernet 0/49:3
!
```

5. Display the configuration of mirror attachment under interface.

```
device# show running-config interface ethernet 0/49:1
interface ethernet 0/49:1

    mirror m1 in
    mirror m2 in
    !
device#

device(config-if-eth-0/2)# do show mirror all
Number of mirrors: 1
      Name : mirror1
  Description : -
Source Interface : ethernet 0/2
  Direction : MIRROR_INGRESS
Source Interface : ethernet 0/2
  Direction : MIRROR_EGRESS
Destination Interface : ethernet 0/1

device(config-if-eth-0/2)#
```



Maintenance Mode

[Maintenance Mode Overview](#) on page 119

[Enabling or Disabling Maintenance Mode](#) on page 119

[MLAG and BGP Module Behavior](#) on page 120

[CLI Commands for Maintenance Mode](#) on page 121

[RASLog and SNMP Traps for Maintenance Mode](#) on page 124

[Enter Maintenance Mode Before Performing Device Maintenance](#) on page 126

[Rebooting into Maintenance Mode](#) on page 127

Learn about Maintenance Mode, a feature that isolates a switch from the network when it goes down for maintenance or upgrades, ensuring minimal disruption to network operations.

Maintenance Mode Overview

The Maintenance Mode feature is designed to isolate a switch from the network during maintenance operations, such as image upgrades. When enabled, all front-panel ports are disabled, diverting traffic to the peer MLAG node. This feature should only be enabled on one MLAG peer at a time.

Key Benefits

1. **Network isolation:** The switch is isolated from connected devices, preventing traffic disruption.
2. **Traffic diversion:** Traffic is diverted to the peer MLAG node, ensuring network continuity.

Configuration Options

1. **Manual enablement:** Maintenance mode can be enabled manually on a running switch.
2. **Automatic enablement on reload:** Maintenance mode can be configured to enable automatically when the switch is reloaded. When the switch comes back online, all physical ports will be in a link-down state, isolating the device from the network.

Enabling or Disabling Maintenance Mode

The Interface Manager processes gNOI messages to enable or disable Maintenance Mode on a switch.

Maintenance Mode Enable Sequence

When Maintenance Mode is enabled, the following steps occur:

1. **Notification to MLAG MS:** The Interface Manager notifies MLAG MS to prevent the peer node from entering a split-brain condition.
2. **Wait for peer node processing:** The system waits for 5 seconds to allow the peer node to process the notification.
3. **Disable front-panel ports:** The Interface Manager sends a notification to FWD-HAL to disable all front-panel ports (except internal and RME ports) in a single protobuf message.
4. **Disable NVE loopback interfaces:** All NVE loopback interfaces are disabled.
5. **Traffic diversion:** Connected nodes detect the port operational state and reprogram to move traffic to the other MLAG node.

Maintenance Mode Disable Sequence

When Maintenance Mode is disabled, the following steps occur:

1. **Notification to MLAG MS:** The Interface Manager notifies MLAG MS to enable secondary keepalive detection.
2. **Bring up delay timer:** MLAG MS starts a bring-up delay timer.
3. **Enable front-panel ports:** All front-panel ports (except MLAG client and uplink ports) are enabled.
4. **ISL formation:** Once the ISL is up, the MLAG session is formed.
5. **Uplink and client port enablement:** After the bring-up delay timer expires, MLAG triggers the enablement of uplink and client ports.

Maintenance Mode On Reload

Maintenance Mode can be configured to enable automatically after a switch reload. In this mode:

1. **Physical ports and NVE loopback interfaces are down:** All physical ports and NVE loopback interfaces are in a down state when the switch comes up after reload.
2. **Maintenance mode disable:** Maintenance mode can be disabled after maintenance operations are complete to bring the switch back into operation.

MLAG and BGP Module Behavior

The MLAG and BGP modules interact with the Interface Manager to ensure smooth operation during Maintenance Mode.

MLAG Bring-up Delay

After Maintenance Mode is disabled, the MLAG bring-up delay timer is restarted. During this delay:

1. **MAC/ARP sync and hardware programming:** MLAG completes MAC/ARP sync and hardware programming.

2. **MLAG client and uplink track ports:** After the timer expires, MLAG client and uplink track ports are brought up.

BGP Module Behavior

IGP route filtering: During Maintenance Mode, BGP filters out IGP routes received from the peer MLAG node for NVE IP, preventing the spine from getting ECMP routes when the NVE IP is down.

Restart Handling

The system ensures that the Maintenance Mode state is preserved during restarts:

1. **Interface Manager:** Stores Maintenance Mode state in PSDB and restores it after restart, applying it to front-panel ports and NVE loopback interfaces.
2. **MLAG MS:** Retrieves Maintenance Mode state from SDB after restart and comes to the correct state.
3. **BGP MS:** Comes up to the same state after restart during the Maintenance Mode bring-up delay duration. The bring-up delay value may need to be adjusted to accommodate the BGP restart duration.

CLI Commands for Maintenance Mode

The system provides various CLI commands to manage Maintenance Mode.

Operational Commands

1. **Enable/Disable Maintenance Mode:** The **system maintenance-mode enable/disable** command sends a gNOI message to enable or disable Maintenance Mode.

```
device(config)# do system maintenance-mode enable/disable

This CLI will send the GNOI message to enable/disable maintenance mode

GNOI Message
message MaintenanceModeRequest {
  /*
    true : Enable maintenance mode
    false : Disable maintenance mode
  */
  bool enable = 1;
}
```

2. **Show Maintenance Mode:** The **show system maintenance-mode** command displays Maintenance Mode information, including status and configuration.

```
device# show system maintenance-mode
Maintenance mode: enabled
Status: in-progress
On Reload: enabled
```

Configuration Commands

1. **Maintenance Mode on Reload:** The **[no] enable-on-reload** command enables or disables Maintenance Mode on reload. This command takes effect when the switch is reloaded.

```
device(config-system-maintenance-mode)# [no] enable-on-reload
rw extr-sys-ext:maintenance-mode

+--rw extr-sys-ext:config
| | +--rw extr-sys-ext:enable-on-reload? oc-yang:bool
. . .
| +--ro extr-sys-ext:state
| | +--rw extr-sys-ext:enable-on-reload? oc-yang:bool
| | +--rw extr-sys-ext:enabled? oc-yang:bool
| | +--rw extr-sys-ext:status? extr-sys-ext:STATUS_TYPE

identity STATUS_TYPE {
  description
    "Base identify type for operational status";
}

identity IN_PROGRESS {
  base STATUS_TYPE;
  description
    "Operation In progress";
}

identity COMPLETED {
  base STATUS_TYPE;
  description
    "Operation Completed";
}
```

2. **Maintenance Mode Status:** The **show running-config system maintenance-mode** command shows the maintenance mode status information.

```
device(config)# show running-config system maintenance-mode
system
  maintenance-mode
  enable-on-reload
!
DUT(config)#

DUT(config)# show running-config system
system
  ssh-server DEFAULT
  vrf mgmt-vrf
  enable
!
  maintenance-mode
  enable-on-reload
!
device(config)#
```

MLAG Commands

1. **Show MLAG Peer:** The **show mlag peer** command displays MLAG peer information, including the peer node's Maintenance Mode state.

```
spine# show mlag peer
Peer towards-leaf
=====
Peer State           : UP
```

```

MCP State           : UP
Role                : BACKUP
Elected MAC        : 00:16:3e:54:e1:00
Extend Bridge Count : 0
Peer Exception     : Peer Under Maintenance Mode, Unhealthy State

```

2. **Debug Commands:** The `curl 0:9004/show-peerdb` and `curl 0:9004/show-peer --data detail=1` commands provide additional information about MLAG peers and the Maintenance Mode state.

```

curl 0:9004/show-peerdb

[admin@leaf]# curl 0:9004/show-peerdb

Dumping MLAG Config Data Structures:
*****

-----
Keepalive Interval    : ---
Keepalive Delay       : ---
Bringup Delay         : 30
Multiplier            : ---
Role                  : ---
Mac                   : ---
Mgmt IP               : 10.38.59.158 Idx: 22000001
System MAC            : 00:16:3e:54:e1:00
BringUpDelayTmrSt     : false
Local MaintenanceMode : Disabled

```

The following debug command displays the Remote Maintenance Mode state:

```

curl 0:9004/show-peer --data detail=1

[admin@leaf]# curl 0:9004/show-peer --data detail=1
=====
Peer table:
=====
PeerName:towards-spine
=====
Mgmt IP: 10.38.59.158
ExtNetInstType: ALL
ExtBrDomains :
Tunnels :
      SIp: 10.2.5.5          DIp: 10.2.5.2          IDX:3100001 VRF: 2 IsIsL:
true
State      : UP, Restart: 2
MM State   : Disabled

```

Show Tech Additions

The **show-if** command output includes additional information about Maintenance Mode in the debug command output, including `Phy Info` which displays Maintenance Mode status and other physical interface information:

```

----- show-if -----
Name:ethernet 0/3
Type:TYPE_PHY
Mode:
IfIndex:0x01000060
MTU: L2 9216 Cfg MTU: L2 0

```

```
Total Number of Lifes:0

Phy Info: Admin State UP Oper State DOWN OperStatusReason MlagShut false
Phy Info: Maintenance Mode true
Phy Info: Oper FEC disabled Oper Speed SPEED_100GB Cfg FEC Cfg Speed
Phy Info: MAC e4:db:ae:5d:18:0c SetIfIndexMapPending 0
Phy Info: Link fault signaling true Link fault State No Fault LocalFaults count0
RemoteFaults count 0
```

RASLog and SNMP Traps for Maintenance Mode

The system generates RASLog messages and SNMP traps to indicate Maintenance Mode status changes. The logs and traps provide valuable information for monitoring and troubleshooting Maintenance Mode operations.

RASLog Messages

Peer Maintenance Mode Enable/Disable: The system generates a LogID 17013 when a peer node's Maintenance Mode is enabled or disabled.

```
LogID:17013 Msg:Maintenance Mode is Enabled on MLAG Peer 10.38.59.158
LogID:17013 Msg:Maintenance Mode is Disabled on MLAG Peer 10.38.59.158
```

SNMP MIBs and Traps

For details on Extreme-defined MIB traps for Maintenance Mode, see *Extreme ONE OS Switching v22.2.0.0 SNMP MIB Reference Guide*.

1. **Convergence Status:** indicates whether the maintenance mode entry/exit transition completed within the expected convergence time (completed(1)) or took longer than expected (timedout(2)).
2. **Reason Code:** indicates the reason for entering/exiting Maintenance Mode, such as user action (userAction(1)) or switch reboot (onSwitchReboot(2)).

SNMP trap daemon output

The SNMP trap daemon output shows the receipt and processing of SNMP traps related to Maintenance Mode. The output includes:

1. **Trap Receipt:** The daemon receives SNMP traps from the network, including Maintenance Mode entry and exit traps.
2. **Trap Processing:** The daemon processes the received traps, extracting relevant information such as trap OIDs and variable bindings.
3. **Trap Details:** The output includes details about the traps, such as:
 - **Trap OID:** The OID of the trap, such as 1.3.6.1.4.1.1916.1.57.0.1 or 1.3.6.1.4.1.1916.1.57.0.2.
 - **SysName:** The system name associated with the trap, such as dutb.
 - **Maintenance Mode Status:** The status of Maintenance Mode, including convergence status and reason codes.

The following example output shows the receipt and processing of Maintenance Mode traps:

```
user@ss2:/etc/snmp$ sudo snmptrapd -f -Le -A -m all -M /usr/share/mibs -Lf /var/log/
snmptrapd.log
Expected "::-=" (♦): At line 38 in /usr/share/mibs/EXTREME-BASE-MIB.mib
Unlinked OID in IF-MIB: ifMIB ::= { mib-2 31 }
Undefined identifier: mib-2 near line 21 of /usr/share/mibs/if.mib
NET-SNMP version 5.9.1
2025-02-06 14:55:28 <UNKNOWN> [UDP: [10.38.61.28]:58598->[10.37.32.18]:169]:
SNMPv2-MIB::sysUpTime.0 = Timeticks: (169100) 0:28:11.00
SNMPv2-MIB::snmpTrapOID.0 = OID: EXTREME-MAINTENANCEMODE-
MIB::extremeMaintenanceModeExitTrap
SNMPv2-MIB::sysName = STRING: dutb EXTREME-MAINTENANCEMODE-
MIB::extremeMaintenanceModeConvergenceStatus = INTEGER: completed(1)
EXTREME-MAINTENANCEMODE-MIB::extreme MaintModeReasonCode = INTEGER: userAction (1)
2025-02-06 14:56:06 <UNKNOWN> [UDP: [10.38.61.28]:58598->[10.37.32.18]:169]:
SNMPv2-MIB::sysUpTime.0 = Timeticks: (169100) 0:28:11.00
SNMPv2-MIB::snmpTrapOID.0 = OID: EXTREME-MAINTENANCEMODE-
MIB::extremeMaintenanceModeEntryTrap
SNMPv2-MIB::sysName = STRING: dutb EXTREME-MAINTENANCEMODE-
MIB::extremeMaintenanceModeConvergenceStatus = INTEGER: completed (1)
MaintModeReasonCode = INTEGER: userAction(1)
EXTREME-MAINTENANCEMODE-MIB::extreme
```

The output also includes the following debug messages from the SNMP agent:

- **Trap Event Received:** The agent receives a trap event and processes it.
- **Trap Data Received:** The agent extracts and processes the trap data.
- **Trap Sent Successfully:** The agent sends the processed trap to the configured destination.

The following is an example of debug message from the SNMP agent:

```
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg":"handleConfig: Notification JSONData: map[system:map[snmp-servers: map[snmp-server:
[map [name:DEFAULT vrfs:map[vrf: [map[hosts:map [host: [map[community-or-user-name:public
config:map[udp-port:169] hostname:10.37.32.18]]] name:mgmt-vrf]]]]]]]]"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg":"handleConfig: snmp vrf config"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg": "handleConfig: vrfName = mgmt-vrf"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg": "handleConfig: snmp trap host"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg":"handleConfig: no update, host config already exists"}
{"Level":"error", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg": "handleConfig: invalid notificaiton"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:16:15.606 UTC +0000",
"Msg":"handleConfig returned error status: <nil>"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:19:15.185 UTC +0000",
"Msg":"Trap event received for trap oid - 1.3.6.1.4.1.1916.1.57.0.2"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:19:15.185 UTC +0000",
"Msg":"Trap Data: Received: &{1.3.6.1.4.1.1916.1.57.0.2 [{dutb 1.3.6.1.2.1.1.5 4} {1
1.3.6.1.4.1.1916.1.57 .1 2} {1 1.3.6.1.4.1.1916.1.57.2_2}]]"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:19:15.190 UTC +0000",
"Msg":"successfully sent trap - 1.3.6.1.4.1.1916.1.57.0.2"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:23:34.418 UTC +0000",
"Msg":"Trap event received for trap oid"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:23:34.418 UTC +0000",
"Msg":"Trap Data: Received: &{1.3.6.1.4.1.1916.1.57.0.1 [{dutb 1.3.6.1.2.1.1.5 4} {1
1.3.6.1.4.1.1916.1.57 .1 2} {1 1.3.6.1.4.1.1916.1.57.2_2}]]"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:23:34.420 UTC +0000",
"Msg":"successfully sent trap 1.3.6.1.4.1.1916.1.57.0.1"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:25:28.465 UTC +0000",
```

```
"Msg":"Trap event received for trap oid - 1.3.6.1.4.1.1916.1.57.0.2"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:25:28.465 UTC +0000",
"Msg":"Trap Data: Received: &{1.3.6.1.4.1.1916.1.57.0.2 [{dutb 1.3.6.1.2.1.1.5 4} {1
1.3.6.1.4.1.1916.1.57 .1 2} {1 1.3.6.1.4.1.1916.1.57.2_2}}]"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:25:28.467 UTC +0000",
"Msg":"successfully sent trap 1.3.6.1.4.1.1916.1.57.0.2"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:26:05.919 UTC +0000",
"Msg":"Trap event received for trap oid - 1.3.6.1.4.1.1916.1.57.0.1"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:26:05.919 UTC
+0000", "Msg":"Trap Data: Received: &{1.3.6.1.4.1.1916.1.57.0.1 [{dutb 1.3.6.1.2.1.1.5
4} {1.1.3.6.1.4.1.1916.1.57 .1 2} {1 1.3.6.1.4.1.1916.1.57.2_2}}]"} -
1.3.6.1.4.1.1916.1.57.0.2"}
{"Level":"debug", "Service": "snmp-agent", "Time":"2025-02-06 09:26:05.921 UTC +0000",
"Msg":"successfully sent trap - 1.3.6.1.4.1.1916.1.57.0.1"}

```

Event Log Messages

Maintenance Mode events are logged, including:

1. Maintenance Mode Enable: logged when Maintenance Mode is enabled.

```
2025-01-30 18:11:48.1094 interface-mgr[7]: Level:info LogID:25028 Msg:Maintenance mode
enabled

```

```
2025-01-30 18:12:14.9646 interface-mgr[7]: Level:info LogID:25026 Msg:Entered
Maintenance mode (Reason: User trigger, Status: COMPLETE)

```

2. Maintenance Mode Disable: logged when Maintenance Mode is disabled.

```
2025-01-30 18:13:10.8355 interface-mgr[7]: Level:info LogID:25029 Msg:Maintenance mode
disabled

```

```
2025-01-30 18:14:42.8429 interface-mgr[7]: Level:info LogID:25027 Msg:Exited
Maintenance mode (Reason: User trigger, Status: COMPLETE)

```

Enter Maintenance Mode Before Performing Device Maintenance

Planned maintenance operations may require the device to be shut down or restarted, resulting in traffic disruption even if alternative paths are available. Maintenance mode provides graceful traffic diversion to alternative traffic paths, helping to minimize traffic loss during such planned operations.

When an alternative path is available, the BGP and MCT protocols redirect traffic away from the node that is going into maintenance mode. When maintenance mode is enabled, all protocols that are running on the maintenance mode node are notified and redirection of traffic (convergence) begins in stages.



Note

Maintenance mode is not supported for the following features: BGP address-family, Flowspec, Layer 3 VPN, VPLS, and VLL (virtual leased line).

1. Access configuration mode.

```
device# configure terminal

```

2. Access system mode.

```
device(config)# system

```

3. Access system maintenance mode.

```
device(config-system)# maintenance-mode

```

4. Enable maintenance mode.

```
ddevice(config-system-maintenance-mode) #
```

The following example enters Maintenance system configuration (config-systemmaintenance-mode) mode:

```
device# configure terminal
device(config)# system
device(config-system)# maintenance-mode
device(config-system-maintenance-mode) #
```

Rebooting into Maintenance Mode

Maintenance mode provides graceful traffic diversion to alternative traffic paths, helping to minimize traffic loss during such planned operations. When an alternative path is available, the BGP and MCT protocols redirect traffic away from the node that is going into maintenance mode. When maintenance mode is enabled, all protocols that are running on the maintenance mode node are notified and redirection of traffic (convergence) begins in stages.

Use the **enable-on-reload** command to enable the device to come up in maintenance mode after a reboot or a reload. This process allows any network errors detected with Extreme ONE OS to be addressed. After the errors have been resolved, the device can be added back to the network.

The following example enables a system reboot or reload into maintenance mode

```
device# configure terminal
device(config)# system
device(config-system)# maintenance-mode
device(config-system-maintenance-mode) # enable-on-reload
device(config-system-maintenance-mode) #
```