



Extreme OS ONE Switching and Routing v22.2.3.0 Management Configuration Guide

Comprehensive Instructions for Configuration and
Management

9041122-00 Rev AA
August 2026



Copyright © 2026 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/>

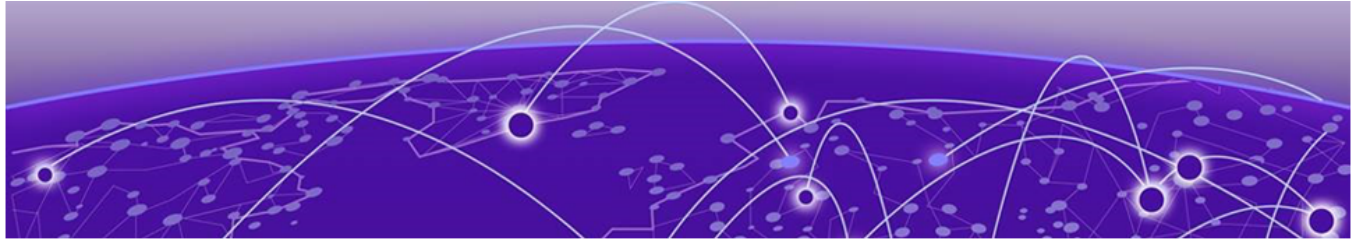


Table of Contents

Abstract.....	ix
Preface.....	x
Text Conventions.....	x
Documentation and Training.....	xi
Open Source Declarations.....	xii
Training.....	xii
Help and Support.....	xii
Subscribe to Product Announcements.....	xiii
Send Feedback.....	xiii
About This Document.....	14
What's New in this Document	15
Supported Platforms.....	19
Ethernet Interfaces and Chassis Devices.....	19
User Management.....	20
Configuration Fundamentals.....	21
Configuration Files.....	21
Default Configuration Files.....	21
Running Configuration Files.....	21
Auto-Persistence of Configuration Data.....	22
Displaying Configurations.....	22
Backing up Configurations.....	24
Configuration File Restoration.....	38
Managing Flash Files.....	39
Copying Support Save Files	40
Session Connection.....	44
Telnet.....	44
SSH.....	46
Configure Terminal Session Parameters.....	50
Configure Login Banner.....	50
Configure MOTD Banner.....	51
System Logging Configuration.....	52
System Logging and Remote Syslog Forwarding.....	53
Supported Features.....	53
Syslog Forwarding to Remote Server.....	54
Local Syslog Messages.....	56
Syslog Local Storage Management.....	57
Printing Syslog Messages on Console and Terminal.....	57
Logging Configurations.....	58
Syslog Message Format (RFC 5424) and Facility Code Support.....	59
Trace Logs (Non-Syslogs).....	60
CLI Commands.....	60

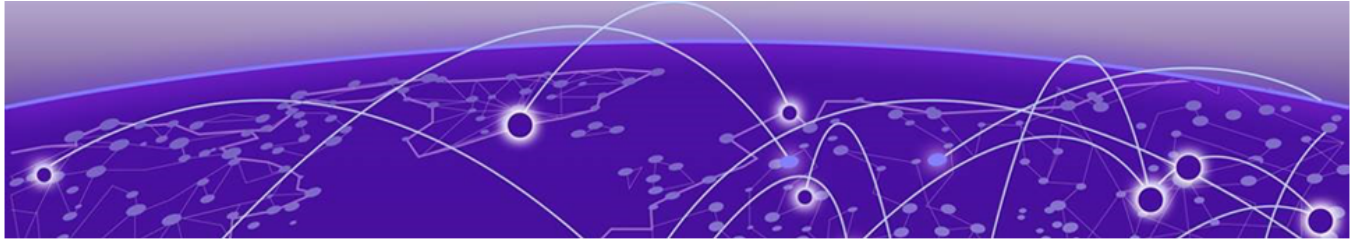
gNMI Commands.....	63
Ethernet Management Interfaces.....	63
Configure Static Management Port.....	64
Display the Management Interface.....	64
Configure an IPv6 Address on the OS ONE Platform.....	68
Port Management.....	69
Extreme OS ONE 100G Ports.....	69
Link Dampening.....	69
Link Fault Signaling.....	75
Dual OOB Redundant Management Port Configuration.....	77
Interface Ethernet Ports.....	78
Displaying Device Interfaces.....	78
System Clock.....	79
Set the Clock.....	79
Management VRFs.....	80
VRF Reachability.....	80
Zero Touch Provisioning (ZTP) over DHCP.....	84
ZTP Configuration.....	84
Configuration and DRC (Drift & Reconcile) Tracking.....	93
Key Features and Capabilities	93
Persistence Handling in CDB vs SDB Reload.....	93
OS ONE and Linux Shell Interoperability.....	95
Admin OS ONE User Capabilities	95
Limitations	95
Access Linux Shell from OS ONE.....	95
Escalate Linux Permissions.....	96
Save and Append Show Command Output to a File.....	97
Logs of Linux Shell Activities.....	97
Firmware Fullinstall Support.....	97
Key Features.....	97
Event Log Messages.....	98
Important Extreme OS ONE Configuration and Certificate Changes.....	98
CLI Commands.....	99
TPVM IAH Extension.....	103
Integrated Appliance Hosting (IAH).....	103
Image Management.....	104
Configure TPVM.....	104
Config CLI.....	106
OS ONE IAH Equivalence.....	106
Exec CLI or gNOI.....	112
OS ONE IAH Equivalence.....	113
Show or gNOI CLI.....	114
OS ONE IAH Equivalence.....	114
RASlog Errors or Logs.....	117
OS ONE IAH Equivalence.....	117
Event Log Messages.....	117
Network Time Protocol and Clock Management.....	119
Network Time Protocol.....	119

NTP Features.....	119
Date and Time Settings.....	119
Time Zone Settings.....	120
Network Time Protocol Server.....	120
Network Time Protocol Client.....	120
Network Time Protocol Associations.....	121
Configure NTP.....	121
Display NTP Server Status.....	122
NTP Server Status When an NTP Server is Not Configured.....	122
NTP Server Status When an NTP Server is Configured.....	123
NTP to Chrony Migration.....	123
NTP vs Chrony.....	123
Limitations.....	123
SNMP Notifications.....	124
Pre-Migration Checklist.....	124
CLI Output: NTP vs Chrony.....	125
SNMP.....	128
How SNMP Monitors Network Devices.....	128
Basic SNMP Operation.....	130
SNMP Community Strings.....	130
SNMP Users.....	130
SNMP Server Hosts.....	131
SNMP Source Interface.....	131
SNMP Engine ID, System Description, Groups, and Views.....	131
Configure SNMP Access Control.....	133
SNMP Informs.....	136
Configure SNMPv2.....	138
Configure SNMPv3.....	139
Supported MIBs.....	140
SNMP Monitoring for Fan and Power Supply Status.....	140
How FRU Monitoring Works.....	141
SNMP Notification Behavior.....	141
MIB Structure and Location.....	142
FRU Status Changed Notification.....	142
Trap Varbind Specifications.....	143
FRU Classification Reference.....	143
FRU Status Values.....	144
Event Logging.....	144
Considerations and Limitations.....	144
Security Considerations.....	145
Monitor Fan and Power Supply Status Using SNMP.....	145
SNMP Monitoring for Optical Transceiver.....	147
Event Log Messages.....	147
Threshold Transitions.....	148
Sample RAS Logs.....	148
SNMP Trap Notification Types.....	148
Monitor Optical Transceivers Using SNMP.....	149
MIB Support for Optical Transceiver.....	152
SNMP Trap Notification Types.....	155

LLDP.....	158
How Network Devices Use LLDP.....	158
LLDP Operation.....	158
LLDP Configuration Scope.....	159
LLDP Timers.....	159
SNMP Support.....	160
Log Locations.....	160
Layer 2 Topology Mapping.....	160
LLDP Configuration Guidelines and Restrictions.....	164
Configuring and Managing LLDP.....	164
Prerequisites.....	164
Supported Platforms.....	165
Limitations.....	165
Understanding the Default LLDP.....	165
Disable LLDP Globally.....	166
Configuring LLDP Global Parameters.....	166
Displaying LLDP Information.....	167
Clear LLDP-Related Information.....	169
DCBx and IEEE LLDP TLVs.....	169
DCBx.....	170
Configure DCBx.....	173
IEEE LLDP TLVs.....	174
LLDP SNMP GET.....	176
Supported SNMP Versions and Operations.....	176
LLDP MIB Support.....	177
Unsupported LLDP Objects.....	177
Limitations.....	177
Retrieving Global LLDP Information.....	177
gNMI and gNOI.....	181
How gNMI and gNOI Protocols Manage Networks.....	181
Troubleshooting Information.....	182
gNMI Protocol.....	182
gNOI Protocol.....	184
gRPC Network Management Interface (gNMI) Service.....	184
Capabilities RPC.....	184
Get RPC.....	185
Set RPC.....	185
Subscribe RPC.....	186
gNMI JSON-IETF Encoding Support.....	187
Use JSON_IETF with Get RPC.....	189
Use JSON_IETF with Subscribe RPC.....	189
Verify Supported Encoding.....	191
Notification Splitting.....	192
JSON_IETF Notification Examples.....	192
gRPC Network Operational Interface (gNOI) Service.....	194
gNOI vs gNMI.....	195
Configure gRPC Server.....	195
Data Model and Northbound Interface.....	196

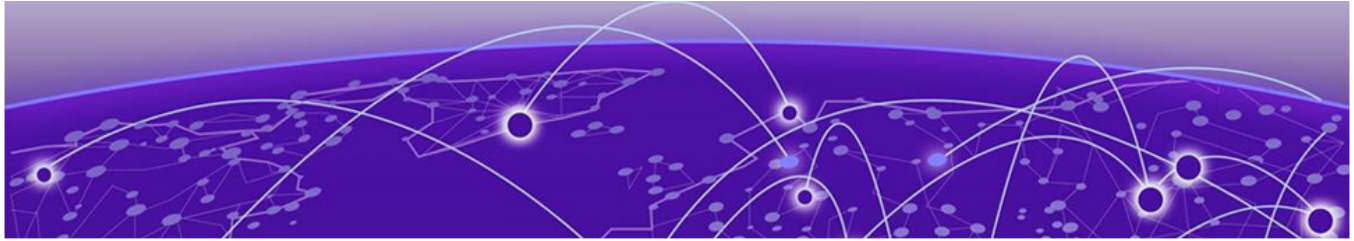
Storage of List Key Values.....	197
Special Characters in Leaf Names and List Key Values.....	197
Printable ASCII Character Support.....	197
Supported Characters.....	199
Module-Specific Restrictions.....	200
Unsupported Values.....	201
Considerations for gNMI Paths.....	201
Inband Management Support.....	201
CLI Commands.....	201
Config Commands.....	201
Exec Commands.....	202
Event Log Messages.....	202
gNMI Authentication and Encryption.....	202
Credentials and Authentication.....	202
Import gNMI Client CA Root Certificate	203
Security.....	203
Encryption.....	203
Authentication.....	204
BMC Configuration.....	205
Increase BMC Security	205
Intelligent Platform Management Interface	205
Baseboard Management Controller	206
Securing BMC	206
Change BMC User Password	206
Enable the BMC Management Interface	206
Configure BMC Management Interface IP Address.....	208
Reset BMC Configuration to Factory Defaults	209
Port Mirroring.....	210
How Mirroring Works.....	210
Types of Mirroring Supported.....	210
Limitations.....	211
Configuration Guidelines.....	211
Configuration Validations for Mirroring.....	211
Single Destination Interface per Mirror Session.....	211
No Port-Channel Members as Destination Interfaces.....	211
No Port-Channel Members as Source Interfaces.....	212
YANG Modules for Mirroring.....	212
Extreme Mirror YANG Module.....	212
OpenConfig ACL YANG Augmentation.....	212
OpenConfig Interfaces YANG Augmentation.....	213
Port Mirroring Configuration Using CLI.....	213
Configure Mirror Destination.....	213
Configure Port-Based Mirroring.....	214
Configure Flow-Based Mirroring.....	214
Use Case Scenarios and Configuration Examples for Port Mirroring.....	214
Local SPAN Port-Based Mirroring.....	214
Local SPAN Flow-Based Mirroring.....	215
Remote Switched Port Analyzer (RSPAN) Mirroring.....	216

RSPAN vs. Local Mirroring.....	217
How RSPAN Works.....	217
Supported Interfaces.....	217
Configuration Restrictions.....	218
Unsupported Operations on Remote-Span VLANs.....	218
Validation Messages.....	219
Configure RSPAN Mirroring.....	219
Maintenance Mode.....	222
How Maintenance Mode Works.....	222
Key Benefits.....	222
Configuration Options.....	222
Enable or Disable Maintenance Mode.....	222
Maintenance Mode Enable Sequence.....	223
Maintenance Mode Disable Sequence.....	223
Maintenance Mode On Reload.....	223
MLAG and BGP Module Behavior.....	223
MLAG Bring-up Delay.....	223
BGP Module Behavior.....	224
CLI Commands for Maintenance Mode.....	224
Operational Commands.....	224
Configuration Commands.....	224
MLAG Commands.....	225
Show Tech Additions.....	226
RASLog and SNMP Traps for Maintenance Mode.....	226
RASLog Messages.....	226
SNMP MIBs and Traps.....	227
Event Log Messages.....	227
Enter Maintenance Mode Before Performing Device Maintenance.....	227
Rebooting into Maintenance Mode	228
License Management.....	229
Why OS ONE Devices Require License.....	229
Key Points.....	229
Self-Authenticated Upgrade (SAU) Licenses.....	229
Deliverables.....	230
OS ONE License Persistence.....	230
CLI Commands and Log Messages for OS ONE License Management.....	230
CLI Commands.....	230
RPCs.....	231
Event Log Messages.....	231



Abstract

This *Extreme OS ONE Switching and Routing Management Configuration Guide* version 22.2.3.0 provides CLI-based instructions for configuring and managing supported platforms. Supported platforms include Extreme 8520/8720/8730/8820. The guide covers essential setup, provisioning, operational workflows, automation, and troubleshooting, with updates for license handling and enhanced management features. Designed for experienced network engineers and administrators, the guide emphasizes secure automation and operational introspection across Extreme OS ONE 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, the product is referred to as *the switch*.

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.

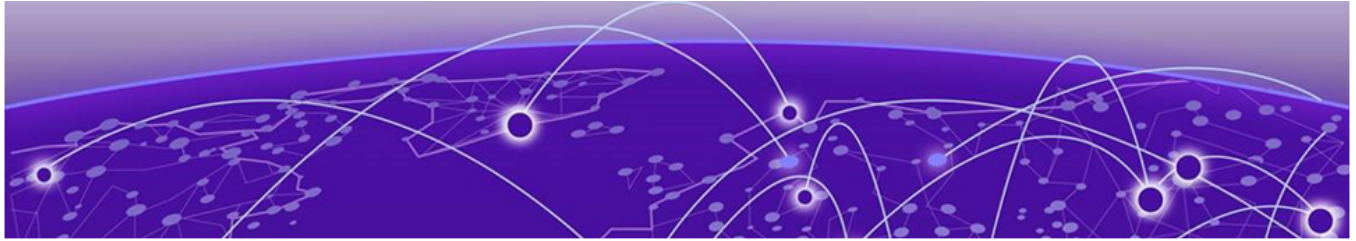
Send Feedback

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 15

[Supported Platforms](#) on page 19

[Ethernet Interfaces and Chassis Devices](#) on page 19

[User Management](#) on page 20

What's New in this Document

The following table describes the information added to this guide for Extreme OS ONE Switching and Routing, release 22.2.3.0:

Feature	Description	Link
SNMP sysDescr, Engine-ID, Groups and Views	Added the following new topics: • SNMP Engine ID, System Description, Groups, and Views: Explains about NMP Engine ID, System Description, Groups, and Views • Configure SNMP Access Control: Describes procedure to configure SNMP access control. • Upgrade and Downgrade Behavior: Describes the system upgrade and downgrade behaviour • Event Log Messages: Describes the event log messages introduced for SNMP Engine ID, System Description, Groups, and Views	• SNMP Engine ID, System Description, Groups, and Views on page 131 • Configure SNMP Access Control on page 133 • Upgrade and Downgrade Behavior on page 135 • Event Log Messages on page 135
Async Operations for Config Replay	The following new topics are added: • Asynchronous Operations for Configuration Replay: Describes the asynchronous operation for config replay. • AsyncOperationService: Describes the AsyncOperationService. • Manage Async Operations: Describes the procedure to manage async operations. • Event Log Messages: Describes the event log messages for asynchronous operations. • Error Handling: Describes the common error scenarios handled	• Asynchronous Operations for Configuration Replay on page 25 • AsyncOperationService Overview on page 26 • Manage Async Operations on page 27 • Event Log Messages on page 27 • Error Handling on page 28

Feature	Description	Link
	by the asynchronous service.	
gNMI JSON-IETF Encoding Support	The following new topics are added: • gNMI JSON-IETF Encoding Support: Describes the gNMI JSON-IETF Encoding Support from gNMI service • Use JSON_IETF with Get RPC: Describes using JSON_IETF with Get RPC • Use JSON_IETF with Subscribe RPC: Describes using JSON_IETF with Subscribe RPC • Verify Supported Encoding: Describes the procedure to verify the JSON_IETF encoding support on target devices • Notification Splitting: Describes the structure when the large subscription response is split across multiple notifications • JSON_IETF Notification Examples: Provides notification examples with and without splitting	• gNMI JSON-IETF Encoding Support on page 187 • Use JSON_IETF with Get RPC on page 189 • Use JSON_IETF with Subscribe RPC on page 189 • Verify Supported Encoding on page 191 • Notification Splitting on page 192 • JSON_IETF Notification Examples on page 192
NTP to Chrony Migration	New topic "NTP to Chrony Migration" describes conceptual information about the migration. New topic "CLI Output: NTP vs Chrony" describes the basic output difference between the NTP and Chrony CLI.	• NTP to Chrony Migration on page 123 • CLI Output: NTP vs Chrony on page 125

Feature	Description	Link
DCBX & Azure Local Extension TLVs	New topics "DCBx and IEEE LLDP TLVs", DCBx, and IEEE LLDP TLVs describe the conceptual information about DCBx and IEEE LLDP TLVs. New topics "Configure DCBx" and "Configure IEEE TLVs" describe the procedure to configure DCBx and IEEE TLVs New topic "DCBx Validation Rules" describes the rules to validate post-configuration.	• DCBx and IEEE LLDP TLVs on page 169 • DCBx on page 170 • Configure DCBx on page 173 • DCBx Validation Rules on page 174 • IEEE LLDP TLVs on page 174 • Configure IEEE TLVs on page 175
SNMP Informs Support	The following new topics are added: SNMP Informs: Provides conceptual information about SNMP Informs. Operational Considerations: Describes upgrade/downgrade behavior, licensing, performance, and security considerations. Configure SNMP Informs: Describes the procedure to configure SNMP Informs.	• SNMP Informs on page 136 • Operational Considerations on page 137 • Configure SNMP Informs on page 137
Syslog with facility code support	The following new topic is added: • Syslog Message Format (RFC 5424) and Facility Code Support: Describes RFC 5424 syslog message format and facility code support The following topics are updated: • System Logging and Remote Syslog Forwarding: Updated the conceptual information with the forwarded message format as RFC 3164 or RFC. 5424 • CLI Commands: Updated the description for the [no] remote-server NAME command	• System Logging and Remote Syslog Forwarding on page 53 • Syslog Message Format (RFC 5424) and Facility Code Support on page 59 • CLI Commands on page 60

Feature	Description	Link
SNMP MIB for Fan and Power Supply and Notification for Failures	The following new topics are added: • SNMP Monitoring for Fan and Power Supply Status: Describes the conceptual information about SNMP monitoring for fan and power supply status. • Monitor Fan and Power Supply Status by Using SNMP: Describes the procedure to monitor fan and power supply status.	• SNMP Monitoring for Fan and Power Supply Status on page 140 • Monitor Fan and Power Supply Status Using SNMP on page 145
SNMP MIB and notification for optical transceivers	The new topics "SNMP Monitoring for Optical Transceiver" and "Monitor Optical Transceivers Using SNMP" provide conceptual information and procedures for monitoring. The new topic "MIB Support for Optical Transceiver" and "SNMP Trap Notification Types" describes supported MIB and trap details.	• SNMP Monitoring for Optical Transceiver on page 147 • Monitor Optical Transceivers Using SNMP on page 149 • MIB Support for Optical Transceiver on page 152 • SNMP Trap Notification Types on page 155
Remove TPVM image after successful deploy	New topic "Image Management" describes that the successful VM import automatically deletes the source image file from the flash storage. Updated the "RASlog Errors or Logs" topic.	• Image Management on page 104 • RASlog Errors or Logs on page 117

Feature	Description	Link
LLDP	The following existing topics are updated: - How Network Devices Use LLDP: Updated with LLDP operation, LLDP configuration scope, LLDP timers, SNMP support, and log locations. - Configuring and Managing LLDP: Updated with prerequisites, supported platforms and limitations. The following new topics are added: - LLDP SNMP GET: Describes the LLDP SNMP GET feature. - Retrieving Global LLDP Information: Provides examples for retrieving global LLDP information.	How Network Devices Use LLDP on page 158 Configuring and Managing LLDP on page 164 LLDP SNMP GET on page 176 Retrieving Global LLDP Information on page 177
RSPAN Mirroring	New topics "Remote Switched Port Analyzer (RSPAN) Mirroring" and "Configure RSPAN Mirroring" describe RSPAN mirroring and its configuration procedure.	Remote Switched Port Analyzer (RSPAN) Mirroring on page 216 Configure RSPAN Mirroring on page 219

For additional information, see the *Extreme OS ONE Switching and Routing Release Notes*.

Supported Platforms

Extreme OS ONE Switching and Routing 22.2.3.0 and later releases support Extreme 8520, Extreme 8720, Extreme 8730, and Extreme 8820 hardware platforms.



Note

Although many software and hardware configurations are tested and supported for this release, all possible configurations and scenarios are beyond this document's scope.

For information about other releases, see the documentation for those releases.

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. Contact your Extreme Networks sales representative for the default password.

During the first login via CLI, you will be prompted to change both the admin password and the GRUB password. You 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, enter the GRUB login credentials (username: root, password: user-set password).

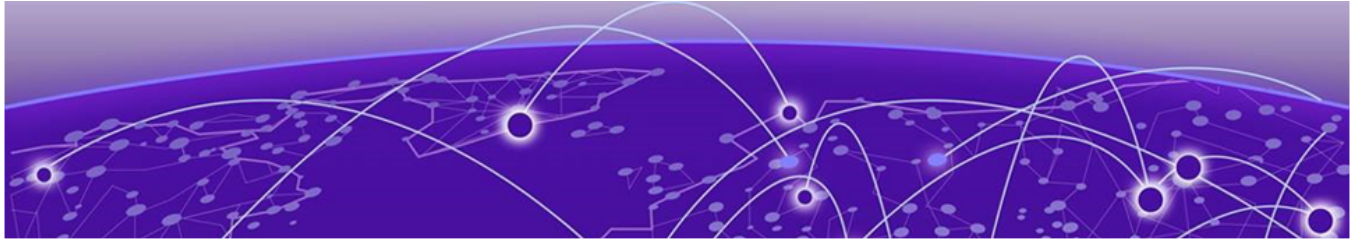
The new password must meet the following criteria:

- 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 21
[Session Connection](#) on page 44
[System Logging Configuration](#) on page 52
[System Logging and Remote Syslog Forwarding](#) on page 53
[Ethernet Management Interfaces](#) on page 63
[Configure an IPv6 Address on the OS ONE Platform](#) on page 68
[Port Management](#) on page 69
[Interface Ethernet Ports](#) on page 78
[System Clock](#) on page 79
[Management VRFs](#) on page 80
[Zero Touch Provisioning \(ZTP\) over DHCP](#) on page 84
[Configuration and DRC \(Drift & Reconcile\) Tracking](#) on page 93

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. The following output is truncated.

```
device# show running-config
! Firmware: 10.2.1.0-20251113_152315.UTC-vwarad, Application(s):
22.2.3.0-20251113_152709.UTC-vwarad
! Date and Time: Sat, 2025-11-22 14:19:40 UTC
system
  grub
    username root password-hashed
grub.pbkdf2.sha512.10000.46FCFD6ED2118C98BC45F9EF4A1963AC4D4304F5E4D72325
8974ED3C1CF6BA910A48355BCCA63C9EC6EFE22479C7C612B2B1661843BF70C95EB726431317B852.E53F87925
9A261EB5E07809ACC
87A87D9E256D5D25A868C7E067A33981551282801023CFB96756AF7569D83C8B288D509A375987529AC4CD245B
9DE41F0A6CE7
!
aaa
  authentication
    admin-user admin password-hashed $6$v7a4pJNPannughVD$SI1M4CRBY3.2s8oDJ5wtGtOKT4m.n4/
aYgudrw8GAvZxn44e
dsNg3QxSTPoeNh9FF6AT29PnOYHFbK4ILbdhi1
!
!
ssh-server DEFAULT
  key rsa 4096
  vrf mgmt-vrf
  enable
!
!
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
    description This is backup
    no shutdown
    !
interface ethernet 0/2
    shutdown
    !
interface ethernet 0/3
    shutdown
    !
interface ethernet 0/4
    shutdown
    !
interface ethernet 0/5
    shutdown
    !
interface ethernet 0/6
    shutdown
    !
interface ethernet 0/7
    shutdown
    !
interface ethernet 0/8
    shutdown
    !
interface ethernet 0/9
    shutdown
    !
interface ethernet 0/10
    shutdown
    !
interface ethernet 0/11
    shutdown
    !
interface ethernet 0/12
    shutdown
    !
interface ethernet 0/13
    shutdown
    !
interface ethernet 0/14
    shutdown
    !
interface ethernet 0/15
    shutdown
    !
interface ethernet 0/16
    shutdown
    !
interface ethernet 0/17
    shutdown
    !
interface ethernet 0/18
    shutdown
    !
interface ethernet 0/19
    shutdown
```

```
!  
interface ethernet 0/20  
    shutdown  
!  
interface ethernet 0/21  
    shutdown  
!  
interface ethernet 0/22  
    shutdown  
!  
interface ethernet 0/23  
    shutdown  
!  
interface ethernet 0/24  
    shutdown  
!  
interface management 0  
    description hi there !  
    ipv4 address dhcp  
    no shutdown  
!  
vrf mgmt-vrf  
    member management 0  
!  
vrf default-vrf  
load-balance ethernet dst-mac src-mac vlan etype  
load-balance ipv4 dst-ip dst-l4-port protocol src-ip src-l4-port  
load-balance ipv6 dst-ip dst-l4-port nxt-hdr src-ip src-l4-port  
control-plane  
!  
device#
```

Backing up Configurations

The Backup Configuration feature ensures that device configurations can be safely backed up and restored, providing an additional layer of reliability and flexibility in network management.

It enables devices to recover their configuration in two key scenarios:

- Automatic recovery on config-db corruption
- Applying a backup configuration on immediate reboot

The key characteristics include:

- **No Default Backup:** By default, there is no backup configuration file on the device.
- **User-Initiated Backup:** A backup configuration is created only when explicitly requested by the user through CLI commands or gNOI calls.
- **Single Backup File:** The system maintains at most one backup configuration file at any given time.
- **Overwriting Existing Backup:** When a new backup configuration is created, it overwrites any existing backup file, ensuring that only the most recent user-created backup configuration is preserved.
- **Plain-Text Format:** The backup configuration file is a plain-text ASCII file, similar in format to the output of the **show running-config** command. The system maintains at most one backup configuration file at any given time.

Asynchronous Operations for Configuration Replay

Release **22.2.3.0** adds asynchronous support for configuration replay operations in Extreme OS ONE SR products. This enhancement introduces **AsyncOperationService**, which allows a configuration replay request to run in the background while the client monitors progress by using an **operation ID**.

This asynchronous workflow is designed for operations that might take time to complete. Instead of waiting for the request to finish in the client session, the system starts the operation, returns an identifier, and continues processing in the background.

The existing synchronous configuration replay RPC remains available and unchanged. The asynchronous workflow is an additional option for clients that require improved tracking and resiliency for long-running operations.

Asynchronous processing provides the following advantages:

- The client does not need to remain blocked while the configuration replay operation runs.
- The system returns an **operation ID** that the client can use to monitor progress.
- Operation state is persisted in the device database, which allows the client to query the operation status after a disconnect or reconnect.
- The service provides visibility into both active and inactive operations through RPCs and CLI commands.

Asynchronous Operation Components:

The following asynchronous operation components are added:

- A shared **AsyncOperationService**.
- A new protobuf definition: **extreme/async_operations/async_operation.proto**.
- An asynchronous **CopyFileToRunning** RPC.
- A **GetOperationStatus** RPC for status retrieval by **operation_id**.
- A **ListOperations** RPC for retrieving tracked operations.

No changes were made to **config_management.proto**, and the existing synchronous RPC remains available.

How Asynchronous Operation Tracking Works

When the client submits an asynchronous **CopyFileToRunning** request, the service accepts the request and returns an **operation ID** and the current operation status. The operation then continues in the background.

The client can use **GetOperationStatus** with the returned **operation_id** to monitor the operation until it reaches a terminal state.

The service tracks operations by using the following states:

- **STARTED** — The operation has been created and initiated.
- **ONGOING** — The operation is currently running.
- **COMPLETED** — The operation finished successfully.

- **FAILED** — The operation ended with an error.
- **ABORTED** — The operation was interrupted before completion.

If the application restarts while an operation is in the **ONGOING** state, the operation transitions to **ABORTED**.

Persistence and Retention Behavior

Operation state is stored in the device database so that the operation can still be tracked after a client disconnects and reconnects. This persistent design improves reliability for long-running asynchronous workflows.

The system retains only the most recent operations. Older completed and failed entries are removed as newer operations are added.

Best Practices

Use the following best practices when working with asynchronous configuration replay:

- Always save the returned **operation_id** immediately after starting an operation.
- Use **GetOperationStatus** to monitor progress until the operation reaches a terminal state.
- If the operation returns **FAILED** or **ABORTED**, review the error details and rerun the request if necessary.
- Use the **show async-operations** CLI commands or **ListOperations** to review the operational history retained by the system.

AsyncOperationService Overview

A new shared service, **AsyncOperationService**, defined in **extreme.async_operation**, provides asynchronous execution support for long-running RPCs. For each RPC that requires asynchronous behavior, the system adds a corresponding asynchronous variant to **AsyncOperationService**. Clients can use **GetOperationStatus** on the same service to monitor operation progress.

CopyFileToRunning RPC

The asynchronous **CopyFileToRunning** RPC accepts the same **CopyConfigRequest** payload as the synchronous **CopyFileToRunning** RPC. Unlike the synchronous implementation, this RPC does not block until the copy operation completes. Instead, the server assigns a unique **operation_id**, starts the copy operation in the background, and streams **OperationStatus** updates to the client in real time. Status transitions follow the operation lifecycle: **STARTED**, **ONGOING**, **COMPLETED**, **FAILED**, or **ABORTED**. The stream closes after the server sends a terminal status.

Clients can also use **GetOperationStatus** with the associated **operation_id** to query the operation state independently at any time. The operation continues to run on the server even if the streaming client disconnects. If the server restarts while the operation is in the **ONGOING** state, the system marks the operation as **ABORTED** during startup and records the reason as **server restarted**. The client receives this updated state the next time it calls **GetOperationStatus**.

GetOperationStatus RPC

The **GetOperationStatus** RPC retrieves the current state of an operation by using its **operation_id**. The response returns an **OperationStatus** object that includes the current state, a human-readable status message, any error details, and the associated start, update, and end timestamps. Supported states are **STARTED**, **ONGOING**, **COMPLETED**, **FAILED**, and **ABORTED**.

If the specified **operation_id** is unknown or has expired, the response returns a **FAILED** state with the message **operation not found or expired**.

ListOperations RPC

The **ListOperations** RPC returns all operations currently tracked by the service and can optionally filter the results by **OperationState**. This RPC is useful for reviewing active operations, auditing recent activity, or troubleshooting failed requests. For each matching operation, the service returns the full **OperationStatus** details.

Manage Async Operations

Follow this procedure to manage async operations.

1. View active operations.

To see operations currently running:

```
show async-operations active
```

This displays all non-completed operations with current state and progress.

2. View operation history.

To see recently completed or failed operations:

```
show async-operations inactive
```

This shows completed, failed, and aborted operations for review.

Event Log Messages

The system generates the following event log messages for asynchronous operations:

Log ID 6021

Message: Async operation '%s' started successfully with operation ID %s

- **Cause:** The asynchronous operation has been initiated.
- **Remedy:** No action is required. Use the operation ID to track progress.
- **Impact:** The operation is running in the background.

Log ID 6022

Message: Async operation '%s' with ID %s completed successfully

- **Cause:** The asynchronous operation completed without errors.
- **Remedy:** No action is required.
- **Impact:** The operation completed as expected.

Log ID 6023

Message: Async operation '%s' with ID %s failed: %s

- **Cause:** The asynchronous operation encountered an error during execution.
- **Remedy:** Review the error details and retry the operation if appropriate.
- **Impact:** The operation did not complete successfully.

Log ID 6024

Message: Async operation '%s' with ID %s was aborted

- **Cause:** The asynchronous operation was interrupted, likely due to a service restart.
- **Remedy:** Retry the operation if needed.
- **Impact:** The operation terminated before completion.

Error Handling

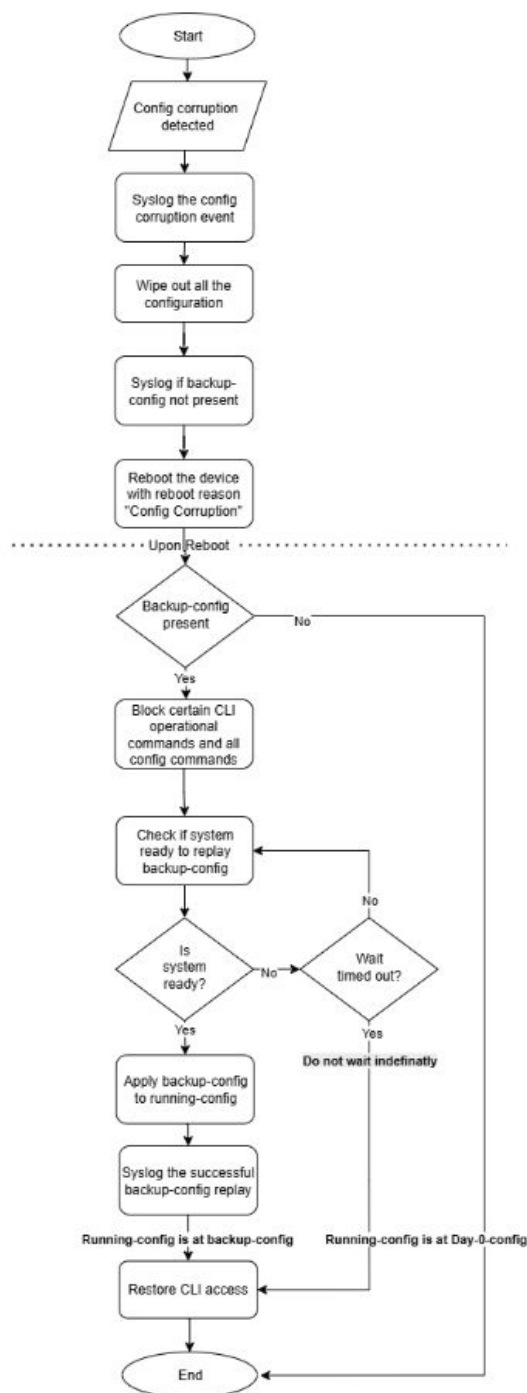
The asynchronous service handles common error scenarios as follows:

- **Client disconnects:** The server continues running the operation in the background. The client can reconnect and query the status later.
- **Server restarts:** Operations in the **ONGOING** state are marked **ABORTED**.
- **CLI command failure during execution:** The operation transitions to **FAILED**, and the service updates the **status_message** with failure information.
- **Invalid GetOperationStatus request:** The service returns a gRPC **InvalidArgument** error.
- **Unknown operation ID:** The service returns a gRPC **NotFound** error.
- **Internal processing failure:** The service returns a gRPC **Internal** error.

Automatic Recovery on Config-DB Corruption

In the event of configuration database (config-db) corruption, the system initiates an automatic recovery process to restore functionality. This automatic recovery process ensures that the system can recover from config-db corruption and resume normal operation with minimal disruption. The following sequence of actions occurs:

1. **Corruption Detection:** The system detects config-db corruption and logs the event.
2. **Configuration Wipeout:** The device performs a complete configuration wipeout and reboots.
3. **Backup Configuration Check:** Upon reboot, the system checks for a backup configuration file
 - **Backup-config present:** The system replays the configuration from the backup-config file during reboot.
 - **No backup-config:** The device boots with the day-0 configuration and the management interface configured for DHCP.
4. **Restricted CLI Access:** During backup-config replay, CLI access is restricted to basic commands such as start-shell, and configuration changes are not allowed.
5. **CLI Access Restoration:** Once the system is fully initialized and the backup configuration is successfully applied, CLI access is restored.



If a backup configuration file exists, the system applies it during the reboot process. If not, the device boots with the day-0 configuration, and the management interface is configured for DHCP.

Applying Backup-Config on Immediate Reboot

To apply a backup configuration on the next reboot, you can use the `apply-on-next-reboot` option when creating the backup configuration file. In contrast to the

Automatic Recovery on Config-DB Corruption scenario, this case does not involve an automatic device reboot. However, when the device reboots for any reason, the user's request to apply backup-config on next reboot is executed. The following sequence of actions occurs:

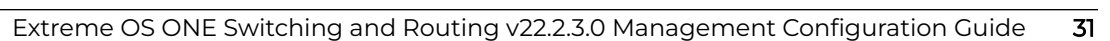
1. **Device Reboot:** The device reboots and recognizes the apply-on-next-reboot request.
2. **Configuration Wipe:** The device wipes out all existing configurations.
3. **Configuration Replay:** The backup configuration is applied, and any commands that fail to apply are ignored and logged.
4. **Temporary CLI Block:** Access to most CLIs is temporarily blocked during the configuration replay
5. **Config Replay Completion:** Once the backup configuration is successfully applied, the config replay is logged, and CLI access is restored.

When applying a backup configuration on the next reboot, consider the following key points:

- The backup configuration will be applied only once on the immediate next reboot.
- Subsequent reboots will not trigger the backup configuration replay again
- The device will wipe out all existing configurations before applying the backup configuration.
- Any commands that fail to apply during the configuration replay will be ignored and logged.
- CLI access will be temporarily blocked during the configuration replay.
- The backup configuration file will be preserved and remain available for future use, including automatic recovery in case of config-db corruption.

Canceling a Pending Application Request

You can cancel an active `apply-on-next-reboot` request by removing the backup configuration file using the **rm backup-config** command or the equivalent gNOI operation. This action disables the pending request, ensuring that the backup configuration is not applied on the next reboot.



Show Backup-Config Replay-Status

Use the **show backup-config replay-status** command to view the current status and progress of applying the backup configuration to the running configuration.

Backup-Config File Persistence

The persistence of the backup configuration file varies depending on the type of system operation. The following table outlines backup configuration file persistence behavior and specifies the conditions under which the backup configuration is applied during various system operations. For instance:

- During full upgrades or rollbacks, the backup configuration is applied only if the user has explicitly triggered `apply-on-next-reboot` beforehand.
- For full installations, the mere presence of a backup configuration is sufficient for it to be applied during the reboot process.

Operation Type	Backup-config File Retained	Backup-config Applied
Firmware upgrade/rollback	Yes	Yes (Only on immediate upgrade/downgrade if 'apply-on-next-reboot' was enabled)
Net/ONIE install	No	NA
Full-Install	Yes	Yes (Mear presence of backup-config file will trigger the config replay)
Full-Install with No-Preserve	No	No
Power cycle / Reload for any reason	Yes	Yes (Only on immediate reboot if 'apply-on-next-reboot' was enabled)
Factory Reset	No	No
Config DB corruption	Yes	Yes

Deliverables and Limitations of Backup Configuration

Use this topic to understand the deliverables and limitations, so you can effectively utilize the Backup Configuration feature and plan for potential constraints.

Deliverables

- **Backup Configuration Feature:** Supports two primary use cases, assuming a backup config file has been saved on the device:
 - Automatic recovery from config-db corruption
 - User-initiated config restore via the `apply-on-next-reboot` option
- **Complete CLI and gNOI Interface:** For managing the backup config feature, including:
 - Creating backup configurations from a file or the running config
 - Displaying backup configuration contents

- Removing backup configurations
- Applying backup configuration to the running-config
- Copying backup configuration to a file

Limitations

- **Manual Backup Creation:** Backup configurations must be created manually by the user. Automatic backup creation or scheduling is not available.
- **Scale Configuration Performance:** Backing up and restoring large-scale configurations can be time and resource-intensive, with duration dependent on configuration size and complexity.
- **CLI Availability During Restore:** CLI access is restricted during backup configuration replay, potentially impacting management operations for large configurations.
- **Single Backup Limitation:** Only one backup configuration file is supported at a time. Creating a new backup will overwrite any existing backup configuration.
- **Validation Requirement:** Backup config files must be validated before saving to the device to ensure configuration integrity.

Commands and Log Messages for Backup Configuration

Use this topic to learn about the commands (CLI and gNOI) and event log messages.

CLI Commands

The following commands are applicable for backup configuration:

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

- **copy backup-config file URL [{vrf VRF-NAME | source-ip ADDRESS}]**

Use this command to copy the backup configuration from disk to a destination file. The destination file can be on the disk, USB or remote location.

- **copy file URI backup-config [apply-on-next-reboot]**

Use this command to copy a configuration file to the backup-config, which can be used to recover from config-db corruption. The apply-on-next-reboot option enables you to apply the backup-config on the next reboot, regardless of the reason for the reboot.

- **copy running-config backup-config [apply-on-next-reboot]**

Use this command to copy the device's running configuration to the backup-config, which can be used to recover from config-db corruption. Additionally, the apply-on-next-reboot option enables you to apply the backup-config on the next reboot, regardless of the reason for the reboot.

- **rm backup-config**

Use this command to remove the backup configuration.

- **show backup-config replay-status**

Use this command to view the current status or progress of automated backup-config replays to running-config (excludes manually triggered config file replays).

- **show backup-config**

Use this command to view the content of backup configuration file.

Show Tech additions

The following logs are available for troubleshooting backup config issues:

- **Backup-Config Replay:** CLI commands executed during backup-config replay are available in the `/var/log/tierra/cli/vtysh_auditlog_admin_admin_LOCAL` log file.
- **gNOI Issues:** For issues related to gNOI, check the `api-gw rastrace` logs located at `/tmp/tierra/trace/api-gw`. Use the **show logging raslogs** and **backup-config** commands.
- **Config Corruption Issues:** For issues related to configuration corruption, check the `config-db rastrace` logs located at `/tmp/tierra/trace/config-db`.

gNOI Commands

```
syntax = "proto3";

package extreme.config_management;

import "extreme/types/types.proto";

option go_package = "github.extremenetworks.com/Engineering/tierra-os/proto/extreme/
config_management";
// Specifies the semantic release version of this proto schema.
option (extreme.types.version) = "0.1.0";

// The ConfigManagement service provides an interface for config related
// operations on Target.
service ConfigManagement {
    // CopyDefaultToRunning RPC is used for copying default-config to
    // running-config. This leads to auto reboot of the device on success.
    rpc CopyDefaultToRunning(CopyDefaultRunningRequest)
        returns (ConfigResponse) {};

    // CopyFileToRunning RPC is used for copying file to running-config.
    rpc CopyFileToRunning(CopyConfigRequest) returns (ConfigResponse) {};

    // CopyRunningToFile RPC is used for copying running-config to a
    // specified configuration file.
    rpc CopyRunningToFile(CopyConfigRequest) returns (ConfigResponse) {};

    // CopyFileToBackupConfig RPC is used to create a backup-config file by
    // copying a config file from various sources such as remote file, USB or
    // disk. Optionally supports apply-on-next-reboot flag (defaults to false)
    // to replay the backup-config on immediate reboot.
    rpc CopyFileToBackupConfig(CopyFileToBackupConfigRequest)
        returns (ConfigResponse) {};

    // CopyRunningConfigToBackupConfig RPC is used to create a backup-config
    // file by copying the current running configuration to the backup config
    // location. Optionally supports apply-on-next-reboot flag (defaults to
    // false) to replay the backup-config on immediate reboot.
    rpc CopyRunningConfigToBackupConfig(CopyRunningConfigToBackupConfigRequest)
```

```

        returns (ConfigResponse) {};

    // RemoveBackupConfig RPC is used to delete the backup-config file and
    // clear any pending apply-on-next-reboot requests.
    rpc RemoveBackupConfig(RemoveBackupConfigRequest)
        returns (ConfigResponse) {};

    // CopyBackupConfigToFile RPC is used to copy the backup-config file to a
    // specified destination file location for export or archival purposes.
    rpc CopyBackupConfigToFile(CopyBackupConfigToFileReq)
        returns (ConfigResponse);
}

message CopyRequest {
    oneof Request {
        CopyDefaultRunningRequest copy_default_running_request = 1;
    }
}

```

Event Log Messages

To enable effective monitoring and troubleshooting of the system, the system utilizes a common logging and event subsystem to:

- Log various messages
- Generate standardized events to indicate:
 - Status changes
 - Errors
 - Failures

The following Security Audit log messages are generated for backup configuration-related events:

- ◦ **LogID 8016:** Backup config created successfully.
- ◦ **LogID 8017:** Backup config created with apply-on-next-reboot.
- ◦ **LogID 8018:** Backup config creation failed.
- ◦ **LogID 8019:** Backup config copied to destination successfully.
- ◦ **LogID 6016:** Backup config restored successfully.
- ◦ **LogID 6017:** Backup config restore failed.
- ◦ **LogID 6018:** Apps not ready within timeout during backup-config replay.
- ◦ **LogID 6019:** Physical interfaces not found during backup-config replay.
- ◦ **LogID 6020:** Indicates that upon reboot due to config corruption, the backup-config file not found, hence default config restored successfully.
- ◦ **LogID 16001:** An unrecoverable configuration database error (config corruption) detected.
- ◦ **LogID 16002:** Config DB corruption, backup-config found, restoring from backup configuration.
- ◦ **LogID 16003:** Config DB corruption, backup-config not found, restoring default configuration
- ◦ **LogID 16004:** Config DB corruption. Failed to reboot the device.

```

{
  "LogID": 8016,
  "Details": {

```

```

        "Level": "Info",
        "Msg": "Backup config created successfully",
        "Cause": "Indicates that backup-config file created successfully",
        "Remedy": "No action required.",
        "Impact": "Backup config created to be used during unfortunate config-corruption."
    }
},
{
    "LogID": 8017,
    "Details": {
        "Level": "Info",
        "Msg": "Backup config created successfully, and it will be applied on next
reboot",
        "Cause": "Indicates that backup-config file created and will be applied on next
reboot or config corruption",
        "Remedy": "No action required.",
        "Impact": "Current configurations will be wiped out and configs from backup-
config file will be applied to running-config on next reboot."
    }
},
{
    "LogID": 8018,
    "Details": {
        "Level": "Error",
        "Msg": "Backup config creation failed",
        "Cause": "Error occurred while creating backup configuration file",
        "Remedy": "Check system resources and try creating backup config again.",
        "Impact": "No backup config available for restoration in case of config
corruption or reboot."
    }
},
{
    "LogID": 6016,
    "Details": {
        "Level": "Info",
        "Msg": "Backup config restored successfully.",
        "Cause": "Indicates that upon reboot, the backup-config is replayed to running-
config successfully.",
        "Remedy": "No action is required.",
        "Impact": "None"
    }
},
{
    "LogID": 6017,
    "Details": {
        "Level": "Error",
        "Msg": "Backup config restore failed.",
        "Cause": "Indicates that upon reboot, the backup-config replay to running-config
failed.",
        "Remedy": "Configure the device to restore the connectivity.",
        "Impact": "Device comes up with Day-0 configuration."
    }
},
{
    "LogID": 6018,
    "Details": {
        "Level": "Error",
        "Msg": "Apps not ready within timeout during backup-config replay.",
        "Cause": "One or more applications did not reach INSTALLED status within the
10-minute timeout period after reboot.",
        "Remedy": "Reconfigure the device if required.",
        "Impact": "Backup-config replay will proceed, but configuration may be partially
applied due to unavailable applications."
    }
}

```

```

    }
  },
  {
    "LogID": 6019,
    "Details": {
      "Level": "Error",
      "Msg": "Physical interfaces not found during backup-config replay.",
      "Cause": "Physical interface are not created within the timeout period after
applications are installed.",
      "Remedy": "Reconfigure the device if required.",
      "Impact": "Backup-config replay will proceed, but configuration may be partially
applied due to unavailable physical interfaces."
    }
  },
  {
    "LogID": 6020,
    "Details": {
      "Level": "Info",
      "Msg": "Default/Day-0 config successfully restored due to config corruption and
backup-config file not available.",
      "Cause": "Indicates that upon reboot due to config corruption, the backup-config
file not found, hence default config restored successfully.",
      "Remedy": "Reconfigure the device.",
      "Impact": "Device now at default/Day-0 configuration."
    }
  }
}
{
  "LogID": 8019,
  "Details": {
    "Level": "Info",
    "Msg": "Backup config copied to destination successfully",
    "Cause": "Indicates that backup-config file copied to destination successfully",
    "Remedy": "No action required.",
    "Impact": "None"
  }
},
{
  "LogID": 16001,
  "Details": {
    "Level": "Error",
    "Msg": "An unrecoverable configuration database error (config corruption)
detected.",
    "Cause": "Configuration database corruption.",
    "Remedy": "No action required. Take a backup of a stable config to avoid
manual config-restore in future",
    "Impact": "Device reboots."
  }
},
{
  "LogID": 16002,
  "Details": {
    "Level": "Error",
    "Msg": "Config DB corruption, backup-config found, restoring from backup
configuration.",
    "Cause": "Configuration DB corrupted and backup-config file found.",
    "Remedy": "No action required.",
    "Impact": "Device reboots and backup-config replayed to running-config."
  }
},
{
  "LogID": 16003,
  "Details": {
    "Level": "Error",
    "Msg": "Config DB corruption, backup-config not found, restoring default

```

```

configuration",
  "Cause": "Configuration DB corrupted and no backup-config file found.",
  "Remedy": "Configure the device to restore the connectivity. And take the
backup of a stable config to avoid manual config-restore in future.",
  "Impact": "Device comes up with default configuration."
},
},
{
  "LogID": 16004,
  "Details": {
    "Level": "Error",
    "Msg": "Config DB corruption. Failed to reboot the device.",
    "Cause": "Failed to reboot the device on Config data corruption.",
    "Remedy": "Manually reboot the device.",
    "Impact": "Device in bad state till rebooted."
  }
}

```

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 (copying a configuration from a disk drive to the current running configuration):

```

device# copy file disk://config-file/runncfg.txt running-config
Applying management interface IP address configuration
..
device#

```

Restore the Default Configuration

Follow this procedure in EXEC mode to restore the default configuration.



Note

To factory default MLAG peers, perform the following steps:

1. Run the **copy default-config running-config** command on the intended MLAG peer.
 2. Enable Maintenance Mode after the device boots up.
 3. Copy the configuration file to the factory default switch.
 4. Disable Maintenance Mode after configuration loading is complete.
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]: y
device#
device# Connection to 10.10.0.1 closed by remote host.
Connection to 10.37.35.1 closed.
```

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.

The following example lists the contents of all directories on the device:

```
device# ls disk://tech-support
tech-support :
-rw----- 6427 2025-04-01 08:54:42 TS-32d-250401T5416.log
-rw-r--r-- 5012653 2025-04-01 08:54:42 TS-32d-250401T5416.tar.gz
-rw----- 6427 2025-04-02 09:13:04 TS-32d-250402T1238.log
-rw-r--r-- 7837214 2025-04-02 09:13:04 TS-32d-250402T1238.tar.gz
device#
```

The following example lists the directories and folders whose names end with the letter 'z' or 'Z' in the disk://tech-support directory:

```
device# ls disk://tech-support | include \wz\b*
-rw-r--r-- 5012653 2025-04-01 08:54:42 TS-32d-250401T5416.tar.gz
-rw-r--r-- 7837214 2025-04-02 09:13:04 TS-32d-250402T1238.tar.gz
device#
```

The following example redirects the listing of the contents of the disk://tech-support directory to a file named disk://config-file/files_for_support_call (and overwrites a file of the same name if one exists):

```
device# ls disk://tech-support | save disk://config-file/files_for_support_call
device#
```

The following example appends the listing of the contents of the disk://coredumps directory to the end of a file named disk://config-file/files_for_support_call:

```
device# ls disk://coredumps | append disk://config-file/files_for_support_call
device#
```

View the Content of a File in Flash Memory

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



Note

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

```
device# cat disk://config-file/testConfig.conf
! Firmware: 10.2.1.0, Application(s): 22.2.3.0
! Date and Time: Mon, 2025-12-10 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/Po5hkbBMNqnSlyPDORUKOELDDdGXsVQ1dRJ.5EksCCnx9dPwTrT.zR1FrCECX1
    !
  !
  grpc-server DEFAULT
    certificate-id ssl-reserved-generated
    mutual-tls
    port 9340
    vrf vrfnew1
  !
  grpc-server gnminew1
    certificate-id test_cert_id
```

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.



Note

You cannot delete a system configuration file in flash memory.

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

Copying Support Save Files

- The following example displays a list of the existing tech support files on the device:

```
device# show tech-support
File Name                               Size (bytes)    Created
TS-dutb-251028T160410.log               8037           2025-10-28 16:04:54
TS-dutb-251028T160410.tar.gz           12298874       2025-10-28 16:04:54
device#
```

- The following example generates a compressed TAR file named TS-48xt-250429T073900.tar.gz and a log file named TS-48xt-250429T073900.log

containing the data to send to Extreme Networks Technical Support for monitoring and troubleshooting. These files are saved locally at `/var/data/disk/techsupport`:

```
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 interface-mgr
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 /run/log/journal
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/base-svc-stdoutlogs
Archiving /var/data/ztp
Archiving /var/log/db-migrator.log
Cleaning up temporary files
TechSupport completed and file TS-48xt-250429T073900.tar.gz generated
Done
device#

```

- The following example generates a compressed TAR file named TS-32d-250620T084936.tar.gz and a log file named TS-32d-250620T084936.log. These files are copied to the /home/user/techsupport/TS/ directory on remote host 1.2.3.4:

```

device# system tech-support scp://exampleuser:examplepassword@1.2.3.4/home/user/
techsupport/TS/
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 interface-mgr
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 /run/log/journal
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/base-svc-stdoutlogs
Archiving /var/data/ztp
Archiving /var/log/db-migrator.log
Cleaning up temporary files
TechSupport completed and file TS-32d-250620T084936.tar.gz generated
Done
Copying the techsupport files to remote server...
The Techsupport is successfully copied to the remote server
device#

```

- The following example displays a list of the existing tech support files on the device:

```

device# ls disk://tech-support
tech-support :
-rw----- 8037      2025-10-28 15:25:39  TS-48xt-250429T073900.log
-rw-r--r-- 11981288 2025-10-28 15:25:39  TS-48xt-250429T073900.tar.gz
device#

```

```

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 tech support 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/

```

The following is an example output of **clear tech-support** command to clear the generated tech support files:

- Without specifying the filename

```

device# clear tech-support
<cr>
FILENAME
device# clear tech-support

```

You can clear either individual tech-support files by specifying the file name or all files at once by not specifying a particular file name. If you choose to clear all files, a warning message will be displayed.

```
device# clear tech-support
WARNING: All the available Techsupport files will be cleared. Do you want to continue?
[y/n]: y
Tech-Support Files cleared
```

- With a specific techsupport filename

```
device# clear tech-support TS-420ax-250922T180231.log
Cleared the Tech-Support file: TS-420ax-250922T180231.log
```

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.

See 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
device(config-system)#
device(config-system)# telnet-server
device(config-system-telnet-server-DEFAULT)#
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.

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

**Note**

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

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.

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

Key Exchange Algorithms

Key exchange algorithms securely exchange a shared session key with a peer. 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

Message Authentication Codes (MACs)

MACs ensure the integrity of messages sent over SSH connections. 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

Encryption Algorithms

Encryption algorithms (ciphers) ensure data privacy over SSH connections. 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

SSH

- Secure Communication: SSH enables secure communication with Extreme OS ONE, running OpenSSH_9.6p1 server. Only the northbound interface is enabled by default in the management VRF.
- Default Configuration: SSH server is enabled by default in the management VRF and is the only northbound interface enabled by default.
- User Authentication: Password-based authentication with local or remote user information storage (TACACS, LDAP, RADIUS).

Configurable SSH Parameters

- Host Key: Manually generate SSH host-key pairs, replacing existing keys. Default: RSA 4096 bits. SSH host keys are preserved in the `/etc/ssh` folder.
- Host Key Preservation: SSH host keys are preserved across firmware upgrades and full installations.
- Supported key types and sizes:
 - RSA: 2048 and 4096 bits (default)
 - ECDSA: 256 bits
- Ciphers, Key Exchange Algorithms, and Message Authentication Codes: Configurable with priority lists.
- Client Alive Interval: Terminate client connection if not reachable within the specified interval (default: 0, disabled).

SSH Server Instances

- Default Instance: Created and associated with the management VRF by default.

- Multiple Instances: Can be configured with unique VRF instances.
- Modification and Deletion: Applicable only to new SSH connections

SSH Server Configuration

Extreme OS ONE 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  
    key rsa 4096  
    vrf mgmt-vrf  
    enable  
  !  
!
```

To check the current running state:

```
device(config-system)# do show running-state system ssh-server  
system  
  ssh-server DEFAULT  
    key rsa 4096  
    vrf mgmt-vrf  
    enable  
  !  
!
```


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:

**Note**

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

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

Connect to an Extreme Device with SSH

Follow this procedure to establish an SSH session to access a device remotely using port 22.

**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.

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 192.0.2.0.

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 OS ONE Software
admin connected from 192.0.2.0 using ssh on device
device#
```

Shut Down and Re-enable 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.

Configure Terminal Session Parameters

Follow this procedure to set the parameters.



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.

Set the timeout length.

The following example configures the terminal timeout to 300 seconds (five minutes) on the device.

```
device# terminal timeout 300
```

Configure 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 UNAUTHORIZED ACCESS TO THIS DEVICE IS NOT ALLOWED.
Activities performed on this device are monitored and recorded. Do not access or
configure this device without explicitly authorized permission. Unauthorized attempts
to
access or configure this device might result in civil or criminal penalties.
```

The banner can be up to 2048 characters long. You can use the **no login-banner** command to remove the banner. For more information about the **login-banner** command, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

3. Verify the configured banner.

```
device(config-system)# do show running-config system banner
system
  login-banner UNAUTHORIZED ACCESS TO THIS DEVICE IS NOT ALLOWED.
  Activities performed on this device are monitored and recorded. Do not access or
  configure this device without explicitly authorized permission. Unauthorized attempts
  to
  access or configure this device might result in civil or criminal penalties.
  !
device(config-system)#
```

The following example is the configuration of the previous steps.

```
device# configure terminal
device(config)# system
device(config-system)# login-banner NAUTHORIZED ACCESS TO THIS DEVICE IS NOT ALLOWED.
Activities performed on this device are monitored and recorded. Do not access or
configure this device without explicitly authorized permission. Unauthorized attempts to
access or configure this device might result in civil or criminal penalties.
```

Configure 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 UNAUTHORIZED ACCESS TO THIS DEVICE IS NOT ALLOWED.
Activities performed
on this device are monitored and recorded. Do not access or configure this device
without explicitly authorized permission. Unauthorized attempts to access or configure
this device might result in civil or criminal penalties.
```

The banner can be up to 2048 characters long. You can use the **no motd-banner** command to remove the banner. For more information about the **motd-banner** command, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

3. Verify the configured banner.

```
device(config-system)# do show running-config system banner
system
  login-banner hellow
  motd-banner UNAUTHORIZED ACCESS TO THIS DEVICE IS NOT ALLOWED. Activities performed
  on this device are monitored and recorded. Do not access or configure this device
  without explicitly authorized permission. Unauthorized attempts to access or configure
  this device might result in civil or criminal penalties.
```

System Logging Configuration

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

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

Command	Description
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, and VRF. The following example displays the remote syslog server configuration that is running currently on the device. In this example, the server uses a VRF instance named red-vrf: <pre> device# show running-config system logging remote-server system logging remote-server 1.1.1.1 mode-transport tcp remote-port 525 secure-forwarding tls vrf red-vrf ! ! device# </pre>
Display Logging Information: - show logging raslog - show system logging facility - show system logging remote-server - show logging audit config firmware security	- 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.

Command	Description
Monitoring: monitor <start stop>	Start or stop displaying Syslog messages on SSH-connected terminal (if enabled). You must run this command in the privileged EXEC mode.
show running-config system logging For example, device# show running-config system logging console On Console session facility Facility information monitor On Monitor session remote-server Remote-server information device#	Display the current logging configuration, including: • Console session facility information • Monitor session information • Remote server information

System Logging and Remote Syslog Forwarding

Extreme OS ONE supports local logging and forwarding of syslog messages to one or more remote syslog servers by using the Linux rsyslog daemon running on the system. The logging framework also supports local storage management, persistent configuration across reloads and firmware upgrades, and optional display of syslog messages on the console or on an SSH-connected terminal.

The platform supports remote syslog forwarding over UDP, TCP, and RELP, and it supports secure forwarding with TLS where applicable. In 22.2.3.0, you can also configure the forwarded message format as RFC 3164 or RFC 5424 on a per-remote-server basis.

Extreme OS ONE logs syslog messages with LogID, level, and description. SysLog messages are forwarded to remote servers using UDP/TCP transport. Each service generates logs.



Note

Device boot may show '[FAILED] Failed to start Rotate log files on console.

```
2025-02-24 06:51:16.8518 chassis-mgr[7]: {"Level":"info","LogID":5016,"Msg":"System initialization is complete. All services are UP"}
```

```
2025-02-17 08:42:05.3090 interface-mgr[8]: {"Level":"info","LogID":25013,"Msg":"Interface ethernet 0/2 Operationally UP"}
```

```
2025-02-17 08:42:52.3133 interface-mgr[8]: {"Level":"info","LogID":25014,"Msg":"Interface ethernet 0/1 Operationally DOWN"}
```

Supported Features

- Syslog Forwarding to remote-server
- Syslog Forwarding over TCP, RELP, RELP + TLS
- Syslog messages local storage management

- Printing Syslog messages on Console, Terminal
- Logging configurations

Syslog Forwarding to Remote Server

The Extreme OS ONE enables remote syslog forwarding via the rsyslogd daemon (version 8.2402.0). This feature allows replication and forwarding of system logs (Syslogs) to one or more configured remote syslog servers using UDP/TCP transport.

Configuration Flow

For detailed information on syslog configuration with TLS, see *Extreme OS ONE Switching and Routing Security Configuration Guide*.

1. User configures remote servers via CLI.
2. CLI Plugin publishes config notifications to Message Bus.
3. Chassis manager Microservice subscribes to Message Bus and processes updates.
4. Chassis manager passes remote server details to OS ONE Service over gRPC.
5. OS ONE service applies configurations to rsyslog daemon via `/etc/rsyslog.d/90-User.conf`.
6. Rsyslog daemon forwards Syslogs to configured remote servers.
7. OS ONE Service acknowledges Chassis-mgr Microservice on successful configuration.
8. Chassis-mgr updates Operational State DB in mgmt-service.
9. Show commands fetch data from Operational State DB.
10. On reload, Chassis-mgr reconciles with OS ONE service over gRPC.

Syslog Forwarding over TCP

- Forwards Syslogs to remote-server over TCP
- Uses TCP protocol for transport
- With default configuration, syslog forwarding uses UDP protocol for transport

Syslog Forwarding over RELP

- Forwards Syslogs over TCP with RELP (Reliable Event Logging Protocol) for reliable delivery (no message loss)

Syslog Forwarding over RELP + TLS

- Forwards Syslogs over TCP with RELP and secure TLS connection
- Ensures reliable delivery (no message loss) with secure connection
- Uses default RELP protocol with TLS encryption

Rsyslog Configurations for Syslog Forwarding

The following are the examples of system logging configurations:

Example 1: TCP with TLS

- Configuration: logging remote-server 10.32.170.30 secure-forwarding tls mode-transport tcp remote-port 525 tls-profile-id tls
- Rsyslog config: /etc/rsyslog.d/90-user.conf

The logging configuration for TCP with TLS generates the following system Rsyslog configuration (/etc/rsyslog.d/90-user.conf):

```
[root@kvm-x86-64]# cat /etc/rsyslog.d/90-user.conf
global(DefaultNetstreamDriverCAFile="/etc/rsyslog.d/ca-list.pem")
action(type="omfwd" Target="10.32.170.30" port="525" protocol="tcp"
StreamDriver="gtls" StreamDriverMode="1" StreamDriverAuthMode="x509/certvalid"
gnutlsPriorityString="NORMAL:-VERS-SSL3.0:-VERS-TLS1.0:-VERS-TLS1.1")
```

Example 2: RELP with TLS

- Configuration: logging remote-server 10.32.170.30 secure-forwarding tls mode-transport relp remote-port 525 tls-profile-id tls
- Rsyslog config: /etc/rsyslog.d/90-user.conf

The logging configuration for RELP with TLS generates the following system Rsyslog configuration (/etc/rsyslog.d/90-user.conf):

```
[root@device]# cat /etc/rsyslog.d/90-user.conf
module(load="omrelp" tls.tlslib="openssl")
action(type="omrelp"
        target="10.32.170.30" port="525" tls="on"
        tls.cacert="/etc/rsyslog.d/ca-list.pem" tls.authmode="certvalid"
        tls.tlsconfcmd="Protocol=ALL,-SSLv2,-SSLv3,-TLSv1,-TLSv1.1")
[root@device]#
```

*Syslog Forwarding Support for VRF and Source-Interface***22.2.1.x Release Support**

- Supports TCP, RELP, TCP+TLS, TCP+VRF, and RELP+TLS
- RELP limitations: No VRF support due to omrelp module constraints

22.2.2.x and Later Release Support

- All combinations supported, including VRF and source-interface
- Source-interface: Uses IP from StateDB path /interfaces/interface[name=*] / subinterfaces/subinterface

- VRF and source-interface config: Patched rsyslog and librelp modules

Transport	VRF	Source-Interface	Example (Rsyslog configuration params)
UDP	Supported	Supported	<pre> action(type="omfwd" protocol="udp" target=" log.example.com " port="514" Device="vrf-blue" Address="192.0.2.10") (Device= binds to VRF; Address= (local IP) supplies source- interface address) </pre>
TCP	Supported	Supported	<pre> action(type="omfwd" protocol="tcp" target="log.example.com" port="514") and ** </pre>
TCP + TLS	Supported	Supported	<pre> action(type="omfwd" protocol="tcp" target="log.example.com" port="6514" StreamDriver="gtls" StreamDriverMode="1" StreamDriverAuthMode="anon") and ** </pre>
RELp	Supported	Supported	<pre> module(load="omrelp") then action(type="omrelp" target="log.example.com" port="20514") and ** </pre>
RELp + TLS	Supported	Supported	<pre> module(load="omrelp") action(type="omrelp" target="log.example.com" port="20515" tls="on" tls.caCert="/etc/rsyslog.d/ ca.pem") and ** </pre>

VRF and Source-Interface Configuration



Note

- omrelp module doesn't support source-interface or VRF natively; yocto patches applied.
- TCP connection source and VRF device binding require patches to rsyslog and librelp modules.

Local Syslog Messages

Syslog messages (RASLogs) to the local system are generated by various system services and filtered into respective microservice directories under `/var/log/tierra/`.

Filter Configuration

The filter is configured in `/etc/rsyslog.d/95-filters.conf`. For example,

```
programname,contains,"chassis-mgr" /var/log/raslogs.log
:programname,contains,"chassis-mgr" /var/log/tierra/chassis-mgr/chassis-mgr.log
& stop
```

Log Storage

- Microservice logs are stored in `/var/log/tierra/<microservice>/`.
- All RASLogs are also written to `/var/log/raslogs.log` for console/terminal display (when enabled).

Syslog Local Storage Management

- Log files are monitored by `logrotated` daemon for size growth
- Check frequency:
 - Every hour if `/var/log` disk space usage < 50%
 - Every 5 minutes if `/var/log` disk space usage > 50%
- Trace messages logged every 5 minutes (for example, `/tmp/tierra/trace/tierra-svc/tierra-svc-RASTrace.log:2499: {"Level":"debug","Service":"tierra-svc","Time":"2025-02-24 13:05:06.884 UTC +0000","Msg":"/var/log disk used: 0.35 GB and free:16.37 GB"} }`)

Log Rotation Configuration

- Config files: `/etc/logrotate.d/` (for example, `/etc/logrotate.d/rsyslog`)
- Log files are stored in the `/var/log/tierra/<service>` directory:
 - Max size: 20MB
 - Max rotations: 5

Journald Configuration

- `systemd/journald.conf` settings:
 - `SystemMaxFileSize=1M`
 - `SystemMaxFiles=5`
 - `RateLimitIntervalSec=3`
 - `RateLimitBurst=10000`
- Limits journal log size and rate to prevent excessive logging

Printing Syslog Messages on Console and Terminal

RASLogs are logged to `/var/log/tierra/<service-name>` and `/var/log/raslogs.log` via the Rsyslog server.

Displaying RASLogs

- Use the `/var/log/raslogs.log` file to display RASLogs on console or terminal.
- The logrotate daemon config manages the log file size.
- A Go routine reads the file (tail order) and filters RASLogs based on level, LogID, and so on.
- Routine stops when feature is disabled.

Console Logging

- Mgmt-cli communicates with Onlp-svc to enable feature.
- Onlp-svc starts Go routine and prints logs to `/dev/console`.

```
device (config)# system
device (config-system) # logging
device (config-system-logging) # console severity info
device (config-system-logging) # 2025-12-05 08:31:28.6679 monitor-svc [3702]: Level:warn
LogID: 27006 Topic:16 Msg:Warning: cpu High Resource usage detected Current usage
Percent: 32.2 2025-12-05 08:31:30.7575 monitor-svc [3702]: Level: info LogID:0 Topic:16
Msg: Detailed System CPU Breakdown System CPU Percent Avg:33 Load Avg 15min:1.34 Load Avg
Percent 5min:33.25 Load Avg Percent 15min:33.5 CPU Count: 4 Load Avg 1min:1.47 Load Avg
5min:1.33 Load Avg Percent 1min: 36.75
2025-12-05 08:31:30.7577 monitor-svc [3702]: Level:info
LogID: 27006 Topic:16 Msg:{"system_cpu": {"per_cpu_percent":
[18.02,7.07,6.93,100], "avg_percent":33,"load_avg_1min":1.47,"load_avg_5mi
n":1.33, "load_avg_15min":1.34,"load_avg_percent_1min":36.75,
"load_avg_percent_5min":33.25, "load_avg_percent_15min":33.5,"cpu_count":4},
"top_processes":[{"name":"vpp_main", "percent":99.88 },{"name":"k35-server",
"percent":7.71},{ "name":"monitor-svc","percent":5.79},{ "name":"ifmgr","percent": 3.84},
{"name":"bgp","percent": 1.9}], "load_status":"low"} 2025-12-05 08:31:32.8727 monitor-svc
[3702]: Level:info LogID:0 Topic:16 Msg: Detailed System CPU Breakdown CPU Count: 4
System CPU Percent Avg:31.43 Load Avg 5min:1.33 Load Avg Percent 1min:36.75 Load Avg
Percent 5min:33.25 Load Avg Percent 15min:33.5 Load Avg 1min:1.47 Load Avg 15min:1.34
```

Terminal Logging

- Mgmt-cli enables feature directly
- Mgmt-cli starts Go routine and prints logs to `stdout` (terminal)

Logging Configurations

- Configure severity level for Syslog messages in each service (default: info)
- Enable/disable console display of RASLogs on console or SSH terminal (default: disabled)
 - When enabled without severity, default error level logs are displayed
- Configure remote-server hosts for Syslog message forwarding
 - IP address, transport protocol (UDP/TCP), port, secure forwarding, vrf, tls-profile-id, and mutual-tls

Configuration Management

- Persistent config changes stored in Mgmt-Service Conf-DB
 - Displayed via Running-Config commands

- Operational and configured remote servers stored in Mgmt-Service State-DB
 - Displayed via Show commands
- Supports restart-ability and reconciles with Mgmt-Service Conf-DB

OpenConfig Compliance

- Follows Openconfig-System-Logging Yang model
- Open-config path for logging: /system/logging

Syslog Message Format (RFC 5424) and Facility Code Support

Syslog Message Format — RFC 5424 Support

Extreme OS ONE 22.2.3.0 introduces support for RFC 5424 syslog message format for remote-server forwarding. You can switch to this format per remote server.



Note

- Default format remains RFC 3164 (legacy BSD syslog)
- New option to switch to RFC 5424 with structured data and high-precision timestamps
- CLI command: message-format RFC5424 under remote-server config

The following RFC formats are supported:

- RFC 3164 (default): Timestamp, hostname, message
- RFC 5424: Structured format with facility/severity encoding, message ID

Example RFC 5424 logs:

```
<190>1 2026-06-09T13:27:07+00:00 kvm-x86-64 chassis-mgr 7 - -  
{ "Level": "info", "Audit": "Firmware", "LogID": 5021, "Msg": "Firmware change successful." }  
  
<187>1 2026-06-09T13:27:07+00:00 kvm-x86-64 base-svc 3743 - -  
{ "Level": "error", "Msg": "expectedBMCVersionGet: failed..." }
```

Facility Code Support

The platform infrastructure now supports configuring syslog facility codes on a per-message basis in `eelog2`. By default, all syslog messages generated by `eelog2` use the `LOCAL7` facility (`syslog.LOG_LOCAL7`).

- Added 12 facility flags to `LogOptions` in `eelog2.go`, including:
 - `FacilityKernOpt`
 - `FacilityDaemonOpt`
 - `FacilityAuthOpt`
 - `FacilityLocal0Opt` through `FacilityLocal7Opt`
- Added **facility-specific syslog logger initialization** during startup by using the `facilitySysLoggers` map in `eelogZeroLog.go`.
- Introduced `newSysLoggerWithFacility()`, which combines the **syslog severity** with the **facility code** in accordance with **RFC 5424**.

- Updated the **default syslog behavior** so that all existing syslog loggers — INFO, WARN, ERROR, FATAL, and PANIC — use LOCAL7 unless a different facility is explicitly specified.
- Added support for **per-message facility selection**, which allows services to assign a specific syslog facility when writing log messages.

Trace Logs (Non-Syslogs)

- Debug and trace logs for internal debugging purposes
- Stored in /tmp/tierra/trace/<service> directory as .log files
- Not managed by rsyslog daemon; uses Go packages zerolog (logging) and lumberjack (file rotation)

Log File Management

- File size limit set in config.json (For example, chassis/configs/config.json)
- Example config:

```
"traceFilename": "/tmp/tierra/trace/chassis-mgr/chassis-mgr-RASTrace.log",
"traceMaxSize": 5, // MB
"traceMaxBackups": 3,
"traceMaxAge": 30,
"traceLogLevel": 0,
```

- File rotation occurs when size limit is exceeded

New DError Logging Feature

- DError level logs written to separate file in /tmp/tierra/trace/ directory
- In addition to regular trace log file
- Logging levels:
 - Debug, DebugError/DError, Trace: written to /tmp/tierra/trace/ log file
 - INFO, WARNING, ERROR, and higher severity: written to /var/log/tierra (Syslog/Raslog) and trace logs in /tmp/tierra/trace/

CLI Commands

The following table describes a brief list of logging related commands:

CLI	Description
service [NAME] severity <emergency alert critical error warning notice info debug>	Configure Syslog logging severity level for a service (default: info) Supported services: those listed in show version command output
[no] <console monitor> severity <emergency alert critical error warning notice info debug>	Enable or disable displaying Syslog messages on console or ssh terminal (default: disabled).

CLI	Description
[no] remote-server NAME	To configure a remote server to forward Syslog messages, you can select from few available options. Here's what you can do under the remote-server submode: • Transport mechanisms: You can choose how to send those logs ◦ mode-transport: Pick TCP, UDP, or something else ◦ secure-forwarding: Use TCP with security if needed • Other tweaks: ◦ message-format: Go with RFC3164 or RFC5424 ◦ remote-port: Set the server port ◦ source-interface: Choose where the logs come from ◦ tls-profile-id: Link a TLS profile for security ◦ vrf: Specify the VRF if needed

CLI	Description
show logging raslogs	Configure remote-server to forward the Syslog messages to remote-server, choose transport mechanisms. and the available attributes under submode are: <pre> device(config-system-logging) # remote-server 2.2.2.2 device(config-system-logging- remote-server-2.2.2.2) # end End current mode and change to enable mode exit Exit current mode and down to parent mode list List all supported configuration commands mode-transport Transport method no Negate a command or set its defaults pwd Display current mode quit Exit current session remote-port Remote server port secure-forwarding Secure forwarding over TCP source-interface Source of syslog packets tls-profile-id tls-profile- id that is associated with the remote logging service vrf vrf mutual-tls Enable mutual TLS (X.509 certificate authentication) for the syslog remote-server </pre>
show system logging facility	Display all the Syslog messages logged on the system, including all services, in time order.
show system logging remote-server	Display the configured severity level for each facility (service)
monitor <start stop>	Display remote-server configuration details
show logging audit config firmware security	Start or stop displaying the Syslog messages on ssh connected terminal if enabled 'logging monitor'
<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 all the logging related configurations.

gNMI Commands

Openconfig path `/system/logging/` is extended for console and monitor display config:

- Enable/disable console display
- Enable/disable monitor display

```

module: openconfig-system
  +--rw system
    | +--rw remote-servers
    | | +--rw remote-server* [host]
    | | | +--rw host -> ../config/host
    | | | +--rw config
    | | | | +--rw host? oc-inet:host
    | | | | +--rw source-address? oc-inet:ip-address
    | | | | +--rw network-instance? oc-ni:network-
  instance-ref
    | | | | +--rw remote-port? oc-inet:port-number
    | | | | +--rw tls-profile-id? string
    | | | | +--rw extr-system-logging-ext:enable? boolean
    | | | | +--rw extr-system-logging-ext:transport? syslog-remote-server-
  transport
    | | | | +--rw extr-system-logging-ext:secure-forwarding? syslog-remote-server-
  secure-forwarding
    | | | | +--rw extr-system-logging-ext:mutual-tls? boolean
    | | | | +--rw extr-system-logging-ext:source-interface? string
    | | | | +--ro state
    | | | | | +--ro host? oc-inet:host
    | | | | | +--ro source-address? oc-inet:ip-address
    | | | | | +--ro network-instance? oc-ni:network-
  instance-ref
    | | | | +--ro remote-port? oc-inet:port-number
    | | | | +--ro tls-profile-id? string
    | | | | +--ro extr-system-logging-ext:enable? boolean
    | | | | +--ro extr-system-logging-ext:transport? syslog-remote-server-
  transport
    | | | | +--ro extr-system-logging-ext:secure-forwarding? syslog-remote-server-
  secure-forwarding
    | | | | +--ro extr-system-logging-ext:mutual-tls? boolean
    | | | | +--ro extr-system-logging-ext:source-interface? string
    | | | | +--rw extr-system-logging-ext:ca-cert? string
    | | | | +--rw extr-system-logging-ext:services
    | | | | | +--rw extr-system-logging-ext:service* [facility]
    | | | | | | +--rw extr-system-logging-ext:facility -> ../config/facility
    | | | | | | +--rw extr-system-logging-ext:config
    | | | | | | | +--rw extr-system-logging-ext:facility? identityref
    | | | | | | | +--rw extr-system-logging-ext:severity? syslog-severity
    | | | | | +--ro extr-system-logging-ext:state
    | | | | | | +--ro extr-system-logging-ext:facility? identityref
    | | | | | | +--ro extr-system-logging-ext:severity? syslog-severity

```

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.

Configure Static Management Port

Before configuring a static address, connect to the device via 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-management-0)# no ipv4 address dhcp
device(config-if-management-0)# ipv4 address 10.x.x.81/20
device(config-if-management-0)# ipv6 address 2001:DB8::xbc:y:e61f:13ff:fe67:4b94/32
device(config-if-management-0)# no shutdown
device(config-if-management-0)#
```

Use the ? symbol to display help for a command

4. Configure the static route.



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
```

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

5. Verify the configuration.

```
device# show running-config interface management 0
interface management 0
  ipv4 address 10.24.85.81/20
  ipv6 address 2001:DB8::xbc:832:e61f:yff:fe67:4b94/32
  no shutdown
!
device#
```

Display 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. The example output may differ for various platforms based on the presence of dual nob, for example, 8820-80C, 8720-32C.

```
device# show int management 0
management 0 Admin state UP      Operational state UP
MTU 1514 bytes
Hardware is Ethernet mac address e4:44:e5:08:ac:00
Current Speed 1G, Duplex: Full
Description: Management 0
DHCPv4 Disabled
IPv4 address 10.32.83.87/24
IPv4 gateway 10.32.83.1
DHCPv6 Disabled
IPv6 address fe80::e644:e5ff:fe08:ac00/64
IPv6 address 2606:9fc0:e004:e083:10:32:83:87/64

Redundancy Mode: active-backup (Dual OOB)
    active-path: mgmt 1
    primary-path: mgmt 1
    all-paths: mgmt 2 mgmt 1

Statistics
    LastClear: 0s
Input:
    Total pkts: 2701008
    Discard pkts: 10
    Errors pkts: 0
    CRC Errors: 0
    MCast pkts: 2664267
    Octets: 150889665
Out:
    Total pkts: 17449
    Discard pkts: 0
    Errors pkts: 0
    MCast pkts: 0
    Octets: 2343705
Rate Info:
    Input: 0.004129 Mbits/sec, 7 pkts/sec 1000000000 speed 0.00% of line-
rate
    Output: 0.000651 Mbits/sec, 1 pkts/sec 1000000000 speed 0.00% of line-
rate
device#
```

show interface extMgmt 1/2 (OOB port)

The **show interface extMgmt 1/2** command is specific to the 8730 platform.

```
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#

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    0x10000020    Ethernet 0/1
Eth 0/2      9216    DOWN       DOWN        400G    0x10000040    Ethernet 0/2
device#

```

show sysinfo media detected

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

S/C Description	Qual	Optical	Type	PartNum	SerialNum	Vendor
0/1 100G SR4 QSFP28 100m	Yes	Yes	QSFP28	EQPT1H4SR4UCM100	2218H-50292	EXTREME NETWORKS
0/3 40G Passive DAC QSFP+ 1m	Yes	No	QSFP_PLUS	EQPC40PC010C0100	2246X-50051	EXTREME NETWORKS
0/4 100G Passive DAC QSFP28 1m	Yes	No	QSFP28	EQPC1HPC010C0100	2214A-53627	EXTREME NETWORKS
0/5 400G-DACP-QSFPDD2M	Yes	No	QSFP-DD	EQDC4HPC020C0100	T0062319A-53729	EXTREME NETWORKS
0/7 400G-FR4-QSFPDD2KM	Yes	Yes	QSFP-DD	EQDT4H8FR4FCL100	T0192337E-50030	EXTREME NETWORKS
0/9 100G Passive DAC QSFP28 1m	Yes	No	QSFP28	EQPC1HPC010C0100	2114A-56882	EXTREME NETWORKS
0/10 40G Passive DAC QSFP+ 1m	Yes	No	QSFP_PLUS	EQPC40PC010C0100	2203X-40362	EXTREME NETWORKS
0/11 400G-DACP-QSFPDD1M	Yes	No	QSFP-DD	EQDC4HPC010C0100	T0052319A-53726	EXTREME NETWORKS
0/12 100G DR QSFP28 500M	Yes	Yes	QSFP28	EQPT1H8DR1DCL100	T0012404E-50004	EXTREME NETWORKS
0/13 40G-SR4-QSFP150M, 850nm	Yes	Yes	QSFP_PLUS	EQPT404SR4VCM100	2109H-50069	EXTREME NETWORKS
0/14 100G Passive DAC QSFP28 1m	Yes	No	QSFP28	EQPC1HPC010C0100	2231A-53265	EXTREME NETWORKS
0/15 400G-AOC-QSFPDD5M	Yes	Yes	QSFP-DD	EQDC4HAC050C0100	T0112313E-50002	EXTREME NETWORKS
0/16 100G 4WDM-40 QSFP28 40km	Yes	Yes	QSFP28	EQPT1H4WM4ECL100	2238E-50070	EXTREME NETWORKS
0/17 100G SR4 QSFP28 100m	Yes	Yes	QSFP28	EQPT1H4SR4UCM100	2218H-50396	EXTREME NETWORKS
0/18 400G-FR4-QSFPDD2KM	Yes	Yes	QSFP-DD	EQDT4H8FR4FCL100	T0192343H-50016	EXTREME NETWORKS
0/21 100G 4WDM-20 QSFP28 20km	Yes	Yes	QSFP28	EQPT1H4WM4WCL100	2215E-50007	EXTREME NETWORKS

0/22	Yes	Yes	QSFP28	EQPT1H4LR4LCL100	2235P-00033	EXTREME NETWORKS
100G LR4 QSFP28 10km						
0/23	Yes	Yes	QSFP_PLUS	EQPT404BD2VCL100	2023L-31062	EXTREME NETWORKS
40G-BDSR-QSFP150M, 850/900nm						
0/24	Yes	Yes	QSFP28	EQPT1H4CW4FCL100	2225H-50015	EXTREME NETWORKS
100G CWDM4 QSFP28 2km						
0/25	Yes	Yes	QSFP28	EQPT1H4LR4LCL100	22256-20003	EXTREME NETWORKS
100G LR4 QSFP28 10km						
0/26	Yes	Yes	QSFP28	EQPT1H4WM4LCL100	2211E-50002	EXTREME NETWORKS
100G 4WDM-10 QSFP28 10km						
0/27	Yes	Yes	QSFP28	EQPT1H4ER4ECL1LT	2128E-50031	EXTREME NETWORKS
100G ER4-lite QSFP28 40km						
0/28	Yes	Yes	QSFP_PLUS	EQPT404SR4SCM1NT	2236H-50406	EXTREME NETWORKS
40G-ESR4-QSFP400M-NT						
----- Management Media						

S/C	Qual	Optical	Type	PartNum	SerialNum	Vendor
Description						

extMgmt 1	Yes	No	SFP	FCLF8520P2BTL-EX	2248P-00730	FINISAR CORP.
1G BASE-T SFP 100m						
extMgmt 2	Yes	No	SFP	FCLF8520P2BTL-EX	1720P-00638	FINISAR CORP.
1G BASE-T SFP 100m						
device#						

show sysinfo media interface <extMgmt 1>

The **show sysinfo media interface <extMgmt 1>** command is specific to the 8730 platform.

```
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>

The **show running-config interface <extMgmt 1>** command is specific to the 8730 platform.

```

device# show running-config interface extMgmt 1

interface extMgmt 1
  no shutdown
!
device#

```

Configure an IPv6 Address on the OS ONE Platform

The following are the basic prerequisites for configuring an IPv6 address on the OS ONE platform:

- Ensure the PC is connected to the serial port of the device.
- Ensure the IPv6 network assignment with a netmask and router address is completed. 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# configure terminal
device(config)# interface ethernet 0/1
device(config-if-eth-0/1)# ipv6 address 2001::100/64
device(config-if-eth-0/1)# ipv6 address 2001::101/64 secondary
device(config-if-eth-0/1)# ipv6 address 2002::100/64
device(config-if-eth-0/1)# ipv6 address 2002::101/64 secondary
device(config-if-eth-0/1)# ipv6 address 2003::/64 eui-64
device(config-if-eth-0/1)# ipv6 address fe80::/64 eui-64
device(config-if-eth-0/1)# exit
device(config)# interface ethernet 0/2
device(config-if-eth-0/2)# ipv6 address fe80::f264:26ff:fe5:c805/64
device(config-if-eth-0/2)#

```

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

```

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

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

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
- Support for the various hardware platforms (8720, 8520-48y, 8520-48XT, 8820-80C, and 8820-40C). For more details, see [Supported Platforms](#) on page 19.

Platforms like the 8520-48XT, 8520-48Y, and 8820-40c have single OOB management, whereas platforms such as the 8720, 8820-80c, and 8730 feature dual OOB with failover redundancy support.

Extreme OS ONE 100G Ports

For fixed form factor Extreme OS ONE 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 Dampening

The Ethernet link dampening feature protects your network from instability caused by rapidly flapping interfaces. When a port experiences repeated up/down transitions within a defined time window, this feature automatically disables the port for a specified period to prevent network-wide disruptions.

This feature is available for all front ports on a device and must be configured at the interface level. When you configure ports in breakout mode (4x10G, 4x25G, or 4x100G), the feature operates independently on each breakout lane. Each lane maintains its own flap count and timers.

This feature is not supported on the management port or on virtual ports.

Link Dampening

Link dampening is an important concept, especially in scenarios where network interfaces have intermittent issues that might lead to link flapping. Excessive flapping can cause unnecessary network instability, especially in environments where routing protocols or network devices heavily rely on link status changes.

This feature helps mitigate network instability by delaying or suppressing link state changes temporarily through a "cooling-off" period after a link fails or recovers before it is allowed to transition again.

Why Use This Feature

A rapidly flapping interface creates serious problems in your network:

- Spanning tree recalculations disrupt the network topology.
- Routing protocol churn causes route instability.
- Flushing of MAC tables interrupts the forwarding of traffic.
- A high CPU load impacts the performance of the control plane.

How This Feature Works

The feature operates through a simple three step process:

1. **Monitor:** For each port, the system counts the link state toggles (the up to down, down to up transitions) within the sampling window that you define on that port.
2. **Evaluate:** When transitions exceed your configured toggle threshold, the system disables the port administratively.
3. **Recover:** After your configured wait period expires, the port re-enables automatically (unless you set the wait time to 0).

When the feature triggers, the port shows an operational status of DOWN (PlDown) and generates a log message for tracking.

Best Practices

Follow these recommendations for effective deployment of link dampening:

- **Start with conservative thresholds:** Begin with higher toggle threshold values and adjust them for your network behavior.
- **Use a sampling time of at least 10 seconds** to ensure stable functioning of link dampening.
- **Match the sampling time to your environment:** Use shorter windows (30–60 seconds) in stable networks and longer windows (120–300 seconds) in environments with expected brief instabilities.
- **Set reasonable wait times:** Allow sufficient time to investigate issues. 300–600 seconds typically provides a good balance.
- **Monitor logs regularly:** Review triggered events to identify problematic interfaces and address root causes.
- **Document your settings:** Maintain records of configured thresholds for consistency across your network.

CLI Commands

The following sections describe how to configure link dampening on an interface and how to restart a port that is operationally down because of excessive link flapping.

Configuring Link Dampening on an Interface

By default, link dampening is disabled on every Ethernet port on the device. You can enable the feature and configure the threshold of link flapping to shut down the port. You can also configure how long it remains shut down.

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

```
device# configure terminal
```

2. Access the interface configuration mode for the port that you want to configure.

```
device(config)# interface ethernet 0/1
```

3. Configure dampening on the port.

```
device(config-if-eth 0/1)# link-error-disable toggle-threshold 10 sampling-time 3  
waittime  
300  
device(config-if-eth 0/1)#
```

This command configures three parameters to control the feature behavior:

- **toggle-threshold** sets the number of link state transitions that trigger the port disablement. The range is 1 through 50 transitions.
 - **sampling-time** sets the time window for counting transitions. The range is 1 through 65535 seconds. Selecting an appropriate sampling time is important, because fast flaps are either missed by a fast timer or over-counted by a slow timer.
 - **wait-time** sets how long the port remains disabled (operationally down) before it is enabled automatically. The range is 0 through 65535 seconds. 0 seconds keeps the port down until you re-enable it manually. 300 seconds (5 minutes) gives you time to investigate before the port re-enables.
4. (Optional) Verify the link dampening configuration.

```
device(config-if-eth 0/1)# do show running-config interface ethernet 0/1  
interface ethernet 0/1  
no shutdown  
link-error-disable toggle-threshold 10 sampling-time 3 wait-time 300  
!  
device#
```

The following example configures link dampening on Ethernet interface 0/1. This configuration disables the port if it flaps 10 times within 3 seconds and keeps it disabled for 300 seconds:

```
device# configure terminal  
device(config)# interface ethernet 0/1  
device(config-if-eth 0/1)# link-error-disable toggle-threshold 10 sampling-time 3 wait-  
time 10  
device(config-if-eth 0/1)#
```

The following example displays the configuration that is running currently on the device for all Ethernet interfaces. In this example, link dampening is enabled only on Ethernet interface 0/1:

```
device# show running-config interface ethernet  
  
interface ethernet 0/1  
no shutdown  
link-error-disable toggle-threshold 10 sampling-time 3 wait-time 300
```

```
subinterface vlan 100
  no shutdown
!
!
interface ethernet 0/2
  shutdown
!
interface ethernet 0/3
  shutdown
!
interface ethernet 0/4
  shutdown
!
interface ethernet 0/5
  shutdown
!
interface ethernet 0/6
  shutdown
!
device#
```

The following example displays details about Ethernet interface 0/1. In this example, Link Error Disable: ON shows that link dampening is enabled on the interface:

```
device# show interface ethernet 0/1

ethernet 0/1 Admin state UP, Oper state UP
Interface index is 16777248 (0x1000020)
MTU 9216 bytes
Hardware is Ethernet  mac address 50:0a:9c:97:48:1d
Current Speed  400G
Description: Ethernet 0/1
FEC:
      Mode: RS-FEC
      Uncorrected Blocks: 18
      Corrected Blocks: 55475033
Link Fault Signaling: ON
Link Fault Status: No Fault
Link Error Disable: ON
Auto Negotiation: ON
Carrier Transitions: 7
      LastClear: 0s
Input:
      Total pkts: 1760026664564
      Broadcast pkts: 0
      Discard pkts: 0
      Errors pkts: 0
      FCS Errors: 0
      MCast pkts: 8
      Octets: 962203743127738
      UCast pkts: 1760026664556
      Runt pkts: 0
      CRC Errors: 0
Input Distribution:
      64 byte pkts: 0
      65-127 byte pkts: 482573358216
      128-255 byte pkts: 358308712804
      256-511 byte pkts: 493289637048
      512-1023 byte pkts: 143312386065
      1024-1518 byte pkts: 181249049849
      Jumbo pkts: 101293520582
Out:
      Total pkts: 1486325156801
      Broadcast pkts: 0
```



```

Discard pkts: 0
Errors pkts: 0
MCast pkts: 20485
Octets: 803095001984686
UCast pkts: 1486325136316
device#

```

Manually Recovering an Operationally Down Port

If a port is down because of excessive flapping, you can fix it before the wait time expires in order to give the port a "fresh start."

Triggering of link dampening causes a link to be operationally down with the PldDown (port link dampening down) reason code if the toggle count exceeds the configured toggle threshold within the configured sampling time. Also, a RASlog notification is generated to help you track flapping issues when link dampening is triggered.

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

```
device# configure terminal
```

2. Examine the operational state (Oper-State) of the ports on the device to determine or verify which link is operationally down with the PldDown reason code.

```
32d# 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#
```

3. Access the interface configuration mode for the port that you want to recover.

```
device(config)# interface ethernet 0/1
```

4. Shut down the port.

```
device(config-if-eth 0/1)# shutdown
```

An administrative shutdown clears all dynamic data, including the flap count and reset timers. It also clears the PldDown reason code.

5. Restart the port.

```
device(config-if-eth 0/1)# no shutdown
```

6. (Optional) Verify that the interface is operationally up (Oper state UP) again.

```

device# show interface ethernet 0/1

ethernet 0/1 Admin state UP, Oper state UP
Interface index is 16777248 (0x1000020)
MTU 9216 bytes
Hardware is Ethernet mac address 50:0a:9c:97:48:1b
Current Speed 400G
Description: Ethernet 0/1
FEC:
Mode: RS-FEC

```

```

        Uncorrected Blocks: 0
        Corrected Blocks: 2199
    Link Fault Signaling: ON
        Link Fault Status: No Fault
        Auto Negotiation: ON
        Carrier Transitions: 1
        LastClear: 0s
    Input:
        Total pkts: 0
        Broadcast pkts: 0
        Discard pkts: 0
        Errors pkts: 0
        FCS Errors: 0
        MCast pkts: 0
        Octets: 0
        UCast pkts: 0
        Runt pkts: 0
        CRC Errors: 0
    Input Distribution:
        64 byte pkts: 0
        65-127 byte pkts: 0
        128-255 byte pkts: 0
        256-511 byte pkts: 0
        512-1023 byte pkts: 0
        1024-1518 byte pkts: 0
        Jumbo pkts: 0
    Out:
        Total pkts: 2
        Broadcast pkts: 0
        Discard pkts: 0
        Errors pkts: 0
        MCast pkts: 2
        Octets: 318
        UCast pkts: 0
    Rate Info:
        Input: 0.000000 Mbits/sec, 0 pkts/sec 0.00% of line-rate
        Output: 0.001432 Mbits/sec, 1 pkts/sec 0.00% of line-rate
device#

```

The following example show how to restart a port that is down because of excessive flapping:

```

device# configure terminal
device(config)# interface ethernet 0/1
device(config-if-eth 0/1)# shutdown
device(config-if-eth 0/1)# no shutdown
device(config-if-eth 0/1)#

```

Displaying Link Dampening Status and Configuration

To display the link dampening enabled/disabled status and the dampening configuration for all front ports on a device, use the **show interface ethernet link-error-disable** command.

The following example displays the status and the configuration for link dampening on each port. In this example, link dampening is enabled and configured only on Ethernet port 0/1:

```

device# show interface ethernet link-error-disable

```

Port	Link Error Disable	Toggle-Threshold	Sampling Time	Wait Time	Status	Last Updated
=====	=====	=====	=====	=====	=====	=====

```

=====
Eth 0      Disable      0          0          0          0
2025-10-09T17:52:13+05:30
Eth 0/1    Enable       10         3         300         0
2025-10-09T17:34:24+05:30
Eth 0/2    Disable      0          0          0          0
2025-10-09T17:34:25+05:30
Eth 0/3    Disable      0          0          0          0
2025-10-09T18:01:28+05:30
Eth 0/4    Disable      0          0          0          0
2025-10-09T18:01:28+05:30
Eth 0/5    Disable      0          0          0          0
2025-10-09T18:01:28+05:30
Eth 0/6    Disable      0          0          0          0
2025-10-09T18:01:28+05:30
device#

```

The **show interface ethernet link-error-disable** command displays the following information::

- Port: Name of the Ethernet port.
- Link Error Disable: Indicates whether link dampening is enabled or disabled on the port.
- Toggle-Threshold: Number of times that the link state cycles from up to down and down to up before the wait period is activated.
- Sampling Time: Number of seconds during which the configured toggle threshold can occur before the wait time is activated.
- Wait Time: Number of seconds that the port remains disabled (down) before it becomes enabled. A value of 0 indicates that the port will stay down until an administrative override occurs.
- Status: Status of the port.
- Last Updated: Time of the last port status change.

Troubleshooting

If a port disables unexpectedly:

1. Check the interface status for the PldDown reason code.
2. Review system logs for Link Error Disable messages.
3. Investigate physical layer issues (such as cable quality, connector problems, or transceiver compatibility).
4. Verify that connected devices operate correctly.
5. Consider adjusting thresholds if legitimate traffic patterns trigger the feature.

For persistent issues, collect diagnostic information during a maintenance window when you can safely disable the feature temporarily.

Link Fault Signaling

Extreme OS ONE 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 UP
  Interface index is 16777248 (0x1000020)
  MTU 9216 bytes
  Hardware is Ethernet mac address 50:0a:9c:97:48:1b
  Current Speed 400G
  Description: Ethernet 0/1
  FEC:
    Mode: RS-FEC
    Uncorrected Blocks: 0
    Corrected Blocks: 2199
  Link Fault Signaling: ON
    Link Fault Status: No Fault
    Auto Negotiation: ON
  Carrier Transitions: 1
    LastClear: 0s
Input:
  Total pkts: 0
  Broadcast pkts: 0
  Discard pkts: 0
  Errors pkts: 0
  FCS Errors: 0
  MCast pkts: 0
    Octets: 0
  UCast pkts: 0
  Runt pkts: 0
  CRC Errors: 0
Input Distribution:
  64 byte pkts: 0
  65-127 byte pkts: 0
  128-255 byte pkts: 0
  256-511 byte pkts: 0
  512-1023 byte pkts: 0
  1024-1518 byte pkts: 0
  Jumbo pkts: 0
Out:
  Total pkts: 2
  Broadcast pkts: 0
  Discard pkts: 0
  Errors pkts: 0
  MCast pkts: 2
    Octets: 318
  UCast pkts: 0
Rate Info:
  Input: 0.000000 Mbits/sec, 0 pkts/sec 0.00% of line-rate
  Output: 0.001432 Mbits/sec, 1 pkts/sec 0.00% of line-rate
device#
```

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 1/1	2	2024-09-22T01:37:04Z	1	2024-09-22T01:36:51Z
Eth 1/2	0	NA	0	NA
Eth 1/3	0	NA	0	NA
Eth 1/4	1	2024-09-22T01:36:50Z	1	2024-09-22T01:36:52Z
Eth 1/5	0	NA	0	NA
Eth 1/6	0	NA	0	NA
Eth 1/7	1	2024-09-22T01:36:50Z	1	2024-09-22T01:36:54Z
Eth 1/8	1	2024-09-22T01:36:50Z	1	2024-09-22T01:36:53Z

```
device#
```

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	ON
Eth 0/3	ON
Eth 0/4	ON
Eth 0/5	ON
Eth 0/6	ON
Eth 0/7	ON
Eth 0/8	OFF

```
device#
```

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 the *Extreme OS ONE Switching and Routing Command Reference Guide*.

Limitations

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

When IAH TPVM boots with both external management links down, a software limitation triggers a sync logic issue. This logic syncs the link status to the TAP IAH VM interface, causing a flap. This behavior is currently a 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.

Set 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
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.
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.

Service	VRF-related command	Additional information
TACACS+	server-group tacacs+-server-address Configure user defined vrf: vrf NAME <String: 1-64 character> vrf name	Uses TCP IPv4 or IPv6. The following example associates a VRF instance named red-vrf with the Terminal Access Controller Access-Control System Plus (TACACS+) server host 6.6.6.6: <pre> device# configure terminal device(config)# system device(config-system)# aaa device(config-system-aaa)# server-group tacacs+ device(config-system-aaa- server-group-tacacs+)# server 6.6.6.6 device(config-system- aaa-server-group-tacacs+ server-6.6.6.6)# vrf red-vrf device(config-system- aaa-server-group-tacacs+ server-6.6.6.6)# do show running-config system aaa server-group tacacs+ system aaa server-group tacacs+ server 6.6.6.6 vrf red-vrf ! ! ! ! device# </pre>

Service	VRF-related command	Additional information
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. The following example associates a VRF instance named red-vrf with the Remote Authentication Dial-In User Service (RADIUS) server host 4.4.4.4: <pre> device# configure terminal device(config)# system device(config-system)# aaa device(config-system-aaa)# server-group radius device(config-system-aaa- server-group-radius)# server 4.4.4.4 device(config-system- aaa-server-group-radius- server-4.4.4.4)# vrf red-vrf device(config-system- aaa-server-group-radius- server-4.4.4.4)# do show running-config system aaa server-group radius system aaa server-group radius server 4.4.4.4 vrf red-vrf ! ! ! ! device# </pre>
LDAP	vrf NAME <String: 1-64 character> vrf name	Default vrf is mgmt-vrf . User-defined VRF is optional. The following example associates a VRF instance named red-vrf with the Lightweight Directory Access Protocol (LDAP) server host 7.7.7.20: <pre> device# configure terminal device(config)# system device(config-system)# aaa device(config-system-aaa)# server-group ldap device(config-system-aaa- server-group-ldap)# server 7.7.7.20 device(config-system- aaa-server-group-ldap- server-7.7.7.20)# vrf red-vrf device(config-system- aaa-server-group-ldap- server-7.7.7.20)# do </pre>

Service	VRF-related command	Additional information
		<pre>show running-config system aaa server-group ldap system aaa server-group ldap server 7.7.7.20 vrf red-vrf ! ! ! ! device#</pre>

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 <pre>device(config-system-grpc- server-test)# vrf VRFNAME <String: 1-64 character> vrf name</pre>	By default, the gRPC server is not configured. You need to configure and enable it. Use this command to create a gRPC server instance.
SSH	ssh-server NAME <pre>#config-system-ssh-server- <ssh_name></pre>	By default, ssh-server 'DEFAULT' instance is created and is associated with 'mgmt-vrf'. A user-defined VRF is optional.

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

Service	VRF-related command	Additional information
Telnet	telnet server NAME <code>config-system-telnet-server-<server_name>#</code>	By default, the Telnet service is not configured, and you need to configure and enable it. Use this command 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	<code>config-system-snmp-server-vrf-<vrf-name>-host-<host-name>-community-or-user-name></code>	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.

Zero Touch Provisioning (ZTP) over DHCP

Learn how to configure and implement ZTP over DHCP for Extreme OS ONE 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

Use this topic to learn about enabling Zero Touch Provisioning (ZTP) on devices to provision new Extreme devices in your network automatically, with minimal manual intervention.

DHCP server

DHCP Server version 4.2.4 is tested on Ubuntu 14.04 (Trusty). The `dhcpd.conf` file has the option 66 (TFTP Server Name) and the option 67 (Filename) set for ZTP. Use the option 66 for the FTP server IP address or host name, and the option 67 for the ZTP configuration file path.

When the device starts the DHCP process, it sends the DHCP client ID to the DHCP server to get the IP address and options 66/67. The device then downloads the ZTP configuration file from the FTP server. To set up a different ZTP configuration file for different devices, you can use the DHCP Client ID in the `dhcpd.conf` file. Ensure to restart the `dhcpd` server whenever `dhcpd.conf` is updated.

The following is an example of DHCP option configuration as ISC DHCP Server for option 66, option 67 and option 43:

```
subnet 10.32.129.0 netmask 255.255.255.0 {
  interface ens160;
  #dhcp option 66
  option tftp-server-name "10.22.33.105";
  #option tftp-server-name "eng.test.com";
  #dhcp option 67
  option bootfile-name "config/files/ztp.conf";
  #dhcp option 43
  option vendor-encapsulated-options
  01:00:2A:68:74:74:70:73:3A:2F:2F:31:30:2E:36:34:2E:31:33:2E:31:30:35:2F:61:70:61:63:68:65:
  2D:73:65:6C:66:73:69:67:6E:65:64:2E:63:72:74;
  #option vendor-encapsulated-options 02:00:01:01;
  range 10.32.129.121 10.32.129.250;
  option domain-name-servers 10.14.15.31;
```

Configuring Option 43 in DHCP Server

Option 43 specifies the URL or protocol for downloading certificates or configurations. It supports two types:

- **Type 01: HTTPS URL**
 - Provides the HTTPS URL to fetch the CA certificate for the server mentioned in Option 66.
 - Format: 01 (type) + 00 XX (length) + hex-encoded URL.
 - Example:
01:00:2A:68:74:74:70:73:3A:2F:2F:31:30:2E:36:34:2E:31:33:2E:31:30:35:2F:61:70:61:63:68:65:2D:73:65:6C:66:73:69:67:6E:65:64:2E:63:72:74

Decodes to: <https://10.64.13.105/apache-selfsigned.crt>

- **Type 02: Boolean Value**
 - Determines whether to use HTTPS or not.
 - Format: 02 (type) + 00 01 (length) + 00 (false) or 01 (true)
 - Examples:
 - 02 00 01 00 (use HTTP if HTTPS fails)
 - 02 00 01 01 (use HTTPS only)

e

Behavior Based on Option 43 Configuration

- **Type 01:** Device downloads CA certificate from specified HTTPS URL and ignores certificate validation.
- **Type 02 with value true:** Device uses HTTPS to download from server in Option 66, assuming a valid certificate from a well-known CA.
- **Type 02 with value false:** Device tries HTTPS first, and if it fails, falls back to HTTP.
- **No Option 43:** Device attempts both HTTP and HTTPS.

HTTP server

Apache server version 2.4.18 was installed and tested on Ubuntu 14.04 (Trusty). The HTTP server stores the ZTP configuration file, firmware, device configuration file, or Python script. The location of these configuration files under the HTTP server base directory is flexible.

ZTP configuration script

The ZTP process can run the script to set up the device configuration automatically. For now, only the Python script is supported. The script takes no parameters. The script can automate any command line, including OS ONE and Linux commands, such as the configuration download command, **copy ftp:// . . . running-config**.

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.
- The client ID used in ZTP does not match the one in the DHCP discover packet.

ZTP configuration file definitions

The following table contains the ZTP configuration file definitions.

Variable description	Description
version	Only version 3 is supported.
date	The last modified date.
supported_nos	The release firmware version supporting the ZTP configuration file.
host_client_id, host_end	Host_client_id marks the beginning of the section host_end marks the end. User could set up the device specific section with full DHCP client id or its prefix. For example, host_client_id=EXTREMENETWORKS##8730-32D##AD012350G-00053 host_end
common_begin, common_end	The setting in the section is shared by all devices.
fwdir	Firmware path in the FTP/HTTP server. For example, Fwdir=/fw/ExtremeOneSR-22.2.3.bin. If base directory of the server is Fwdir=/fw/ExtremeOneSR-22.2.3.bin, then the absolute path of firmware in ftp server is located at /var/www/http/fw/ExtremeOneSR-22.2.3.bin.
startup	The path to the device configuration file in the FTP server. If omitted, the device will take the default configuration. The value can be "default" or user configuration file.
scriptcfgflag	The default is 0, when not specified. The meaning of the value is: 0 - only use startup, script is ignored 1 - only use script, startup is ignored
script	The device configuration Python script file.

The following is an example ZTP configuration (ztp.cfg) file:

```
version=3
date=5/26/2025
```

```
supported_nos=22.2.3.0

common_begin
vcsmode=SA
scriptcfgflag=2
startup=startup.cfg
script=script2.py
fwdir=ExtremeOneSR-22.2.3.0.bin
common_end

host_client_id=EXTREMENETWORKS##ABCD##234567
script=script2.py
startup=/config/abc2.conf
host_end

# Model ABC_1256 and SerialNum SNU3333222
host_client_id=EXTREMENETWORKS##ABC_1256##SNU3333222
scriptcfgflag=2
script=/script/def3.py
host_end
```

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.

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/ztpdhcp/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/ztpdhcp/log/pythonZtp.log` location. The following is a sample Python script:

```
#!/usr/bin/python3
import sys
sys.path.insert(1, '/usr/local/tierra/bin/scripts')
from CLI import CLI

CLI('config; system; ntp; server time1.google.com; enable')
CLI('show running-config system ntp; show clock; show ntp association')
CLI('copy file scp://user:password@1.2.3.5:/var/www/html/startup.cfg running-config')
```

These files (Python scripts), downloaded from the HTTPS server, will persist after a firmware upgrade to the version specified in the `fwdir` flag of the ZTP configuration file.

ZTP Invocation

ZTP will get automatically triggered in the following scenarios:

1. Netinstall: After a netinstall, the device 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 device
- Configuring the device management interface

This automated ZTP process simplifies large-scale device 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 or net install 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 device 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.

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 device 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 OS ONE Switching and Routing Command Reference Guide*.

ztp cancel

The **ztp cancel** command stops the ongoing Zero Touch Provisioning (ZTP) process. This is typically used when the network setup is incorrect, preventing ZTP from starting. By canceling ZTP, the device reverts to non-ZTP mode, allowing configuration commands. Note that the cancel request will fail if the process is in certain stages (firmware upgrade, file replay, or script execution). During ZTP process config commands are not allowed. After canceling, you can run the configuration commands.

factory reset

This exec command brings back the device to ZTP mode just as device boots up in factory.

show ztp dhcp status

```
device# show ztp dhcp status
ZTP Status: Completed
Client-id: EXTREMENETWORKS##8730-32D##AD012451G-00007
ZTP Interface: mal
Interface IP: 10.32.129.191
Http Server IP Address: 10.64.13.105
Transfer Mode: HTTP
ZTP Configuration File: config/files/ztp.conf
Switch Configuration File: startup.cfg, /config/files/script_device.py
```

```
Firmware Image: /config/files/ExtremeOneSR-22.2.3.0.bin
Firmware Download State: firmwareInstallCompleted
device#
```

show ztp dhcp logs

The command shows the ZTP logs at CLI.

```
DHCPztpclient# show ztp dhcp logs
2025/01/31 17:25:43.932280 Received dhcption 66:1.2.3.4 and dhcption 67: config/
files/ztp.conf
2025/01/31 17:25:43.932455 Starting ZTP DHCP Process.
2025/01/31 17:25:48.934303 Downloading from URL : https://1.2.3.4/config/files/
ztp.conf.
2025/01/31 17:26:18.947195 Downloading from URL : http://1.2.3.4/config/files/ztp.conf.
2025/01/31 17:26:58.950679 Downloading from URL : https://1.2.3.4/config/files/
ztp.conf.
2025/01/31 17:27:28.954204 Downloading from URL : http://1.2.3.4/config/files/ztp.conf.
2025/01/31 17:28:08.955744 Downloading from URL : https://1.2.3.4/config/files/
ztp.conf.
2025/01/31 17:28:29.227842 Received dhcption 66:10.64.13.105 and dhcption 67:
config/files/ztp.conf
2025/01/31 17:28:29.227954 Starting ZTP DHCP Process.
2025/01/31 17:28:34.235213 Downloading from URL : https://10.64.13.105/config/files/
ztp.conf.
2025/01/31 17:28:34.305942 Downloading from URL : http://10.64.13.105/config/files/
ztp.conf.
2025/01/31 17:28:34.352092 Successfully downloaded ztp configuration file config/files/
ztp.conf to /var/data/ztp/ztp.conf file.
2025/01/31 17:28:34.352344 Downloading from URL : https://10.64.13.105/startup.cfg.
2025/01/31 17:28:35.410149 Downloading from URL : http://10.64.13.105/startup.cfg.
2025/01/31 17:28:35.456341 Successfully downloaded switch startup config file
startup.cfg to /var/data/ztp/switchConfig.conf.
2025/01/31 17:28:35.456400 Downloading from URL : https://10.64.13.105//config/files/
script.py.
2025/01/31 17:28:36.517075 Downloading from URL : http://10.64.13.105//config/files/
script.py.
2025/01/31 17:28:36.563852 Succesfully downloaded python script /config/files/
script.py to /var/data/ztp/script.py.
2025/01/31 17:28:36.563974 Starting to download firmware [ /config/files/
TOS-10.1.3.0-20250131_171456.UTC-rajans.bin ].
2025/01/31 17:28:36.564320 Downloading from URL : https://10.64.13.105//config/files/
TOS-10.1.3.0-20250131_171456.UTC-rajans.bin.
2025/01/31 17:28:37.624186 Downloading from URL : http://10.64.13.105//config/files/
TOS-10.1.3.0-20250131_171456.UTC-rajans.bin.
2025/01/31 17:28:52.247127 Successfully Downloaded firmware.
2025/01/31 17:28:52.247161 Activating new firmware and going for reboot.
2025/01/31 17:30:56.042702 Starting ZTP DHCP Process.
2025/01/31 17:31:01.043006 Device booted after ztp firmware upgrade.
2025/01/31 17:31:01.044796 Running configuration script : startup.cfg.
2025/01/31 17:31:03.331772 Completed executing script : startup.cfg.
2025/01/31 17:31:03.332278 Running configuration python script : /config/files/
script.py.
2025/01/31 17:31:06.351667 Completed executing python script : /config/files/script.py.
2025/01/31 17:31:06.352417 ZTP DHCP Completed.
```

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. Device Startup Configuration Logs: `/var/data/ztp/log/configReplay.log`
4. Consolidated ZTP Logs: `/var/log/ztp/log/ztp.log`

The **show ztp logs** command uses these log files to provide a centralized view of ZTP-related logs.

Configuration 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:

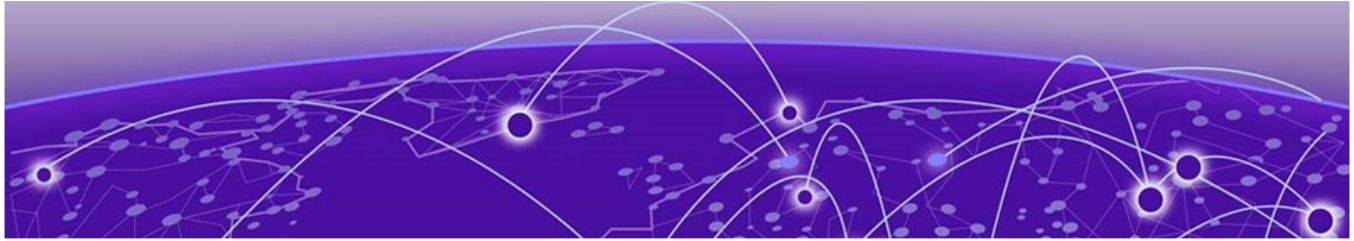
- **Reliable Change Tracking:** Accurately tracks and retrieves metadata for the latest configuration and state updates.
- **Device-Side RPC:** Implements a device-side RPC to fetch the latest version and timestamp for configuration and state changes.
- **Automatic Versioning and Timestamp Updates:** 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.

Scenario	CDB Handling	SDB Handling
Reload / Reboot	After a reboot, the system retrieves the persisted version number from the config database into local memory.	Version resets to 0.
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.



OS ONE and Linux Shell Interoperability

[Admin OS ONE User Capabilities](#) on page 95
[Access Linux Shell from OS ONE](#) on page 95
[Escalate Linux Permissions](#) on page 96
[Save and Append Show Command Output to a File](#) on page 97
[Logs of Linux Shell Activities](#) on page 97
[Firmware Fullinstall Support](#) on page 97

Admin OS ONE User Capabilities

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

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

Limitations

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

Access Linux Shell from OS ONE

Inside the Extreme OS ONE 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, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

1. To access the Extreme OS ONE 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 OS ONE commands from the Linux shell.

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

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

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

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

```
exit
device#
```

Escalate Linux Permissions



Caution

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

Extreme OS ONE 22.2.2.0 and later enforces user awareness and authentication for privileged access in the start-shell using sudo, with a security warning banner and the current admin password of the user required each time.



Note

- Warning banner and password authentication are required for every sudo command.
- The sudo configuration has been updated to disable password caching, require authentication using the current admin password, and display a security banner each time a sudo command is run.
- To avoid the banner and repeated authentication prompts, enter a privileged Bash shell with sudo bash.

Follow this procedure to escalate Linux permissions.

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

```
device# start-shell
[admin@device]# sudo bash
*****
The linux shell access on this device is intended for diagnostics and
debugging purposes solely by the equipment vendor's trained engineers.
Improper use of the functionality made available through the linux shell
access could cause significant harm and disruption to the network operation.

Your use of the functionality made available through the linux shell access
is at your sole risk and you assume all liability resulting from such use.
The equipment vendor shall have no liability for any losses or damages
arising from or relating to linux shell access (and the functionality
enabled thereby) by anyone other than the equipment vendor's
authorized engineers.

Proceeding with the usage of linux shell access on this device explicitly
indicates your agreement to the terms of this disclaimer.
*****
Password:
[root@device]#
```

You can now execute commands that require root permissions.

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

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

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

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

Save and Append Show Command Output to a File

1. Save the **show** command output to either config-file or tech-support file.

```
device# show running-config | append
FILENAME Name of the file
disk://config-file/<file-name>
disk://tech-support/<file-name>
```

2. Run the following command to check the copied file.

```
device# ls
config-file :
-rw-r--r-- 3397 2025-10-31 16:00:19 _backup_
-rw-r--r-- 699868 2025-11-05 13:21:37 mgmsvcs_scale
-rw-r--r-- 1399782 2025-11-07 12:40:24 test.cfg
tech-support :
firmware :
coredumps :
iah :
drwxr-xr-x 4096 2025-10-31 14:43:11 efaboot
device#
```

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

Logs of Linux Shell Activities

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

Firmware Fullinstall Support

This topic outlines the step-by-step process for performing a full, clean install of the provided Extreme OS ONE 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 Extreme OS ONE binary image.

Key Features

- Installation of firmware using the Extreme OS ONE CLI
- Deletion and creation of managed partition

- Replacement of current configuration with the default Extreme OS ONE 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 OS ONE Configuration and Certificate Changes

Extreme OS ONE Configuration: Installing new firmware overwrites your current configuration with the default Extreme OS ONE 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 are 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

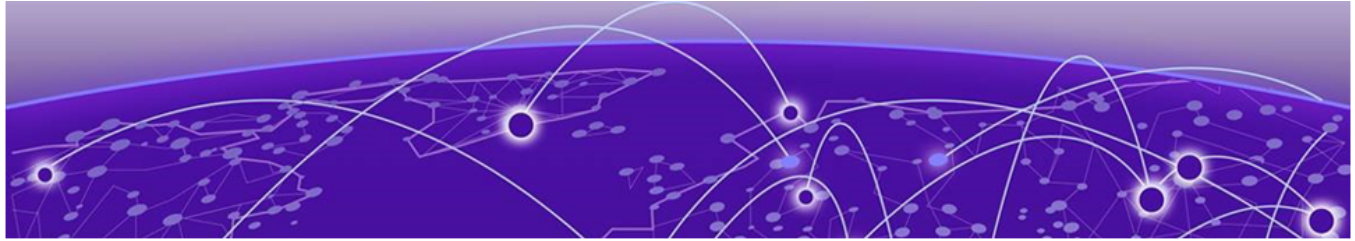
The following table describes CLI commands to complete the system firmware fullinstall:

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 you 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	This command simulates the 'onie-nos-install' process for new Extreme OS ONE firmware. Here's what it does: • Deletes all device partitions • Preserves management certification files by default (unless 'no-preserve' option is used) • Initiates new Extreme OS ONE 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 firmware 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> -OR- http[https]: //[username:password@]<host>[:port]/<filepath> 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	The following openconfig gNOI RPCs are used or supported: <pre>rpc Activate(ActivateRequest) returns (ActivateResponse); rpc Verify(VerifyRequest) returns (VerifyResponse);</pre> The following are the corresponding Request and Response messages: <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/OSOneSR-22.2.3.0-110.bin </pre>



TPVM IAH Extension

[Integrated Appliance Hosting \(IAH\)](#) on page 103

[Image Management](#) on page 104

[Configure TPVM](#) on page 104

[Config CLI](#) on page 106

[Exec CLI or gNOI](#) on page 112

[Show or gNOI CLI](#) on page 114

[RASlog Errors or Logs](#) on page 117

Use this topic to learn about the TPVM IAH Extension.

Integrated Appliance Hosting (IAH)

IAH is a microservice in the Extreme OS ONE 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

Image Management

VM images required for deployment are downloaded and stored on the switch's persistent flash storage at `/var/data/disk/iah/`. This directory functions as the staging repository for all IAH image artifacts, including TPVM packages (`.deb`) and OVA files.

During the import process, the IAH service reads the source package from this location, extracts the compressed QCOW2 disk image, and provisions the VM by creating its virtual disks within a dedicated per-VM image directory. The source package remains in `/var/data/disk/iah/` until the import operation is completed.

By default, upon successful completion of the VM import, the source image file is automatically removed from `/var/data/disk/iah/` to reclaim disk space. This cleanup is important because the package size can be significant (typically around 3 GB for TPVM) and is no longer required once the VM disks are fully extracted and deployed.

If multiple deployments are required from the same image, you can retain the source package by specifying the `nodelete` option in the CLI import command. For example:

```
iah vm import TPVM tpvm disk://iah/tpvm-4.7.15-2.amd64.deb nodelete
```

When the `nodelete` option is used, the system preserves the source image file and logs that the file was not deleted.

In the event of an import failure, the source image is never deleted, regardless of the option used. This behavior ensures that the image remains available for subsequent retry operations without requiring a re-download.

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 the *Extreme OS ONE Switching and Routing 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. Verify VM configuration details.

```
# show iah vm TPVM details
```

6. Switch to configuration mode.

```
# conf terminal
```

7. Access IAH configuration mode.

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

8. Configure the IAH TPVM configuration block.

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

9. Example configuration to update CPU settings.

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

10. Example configuration to update RAM settings.

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

11. Set the hostname.

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

12. Set the timezone.

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

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

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

14. Configure any other supported attributes.

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

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

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

16. Example configuration for editing disk settings.

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

17. Deploy the configuration.

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

18. Set IPv4 address and gateway.

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

19. Change the hostname.

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

20. Add or update NTP servers.

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

21. Update password.

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

22. 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.

OS ONE IAH Equivalence

- 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 OS ONE 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/ipv6 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.

OS ONE also provides implicit password-less access for "extreme@TPVM".

6. Implicit for OS ONE.

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



Note

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

- 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 is automatically created and mounted. If a specific size is required, you can utilize the following method:

- IAH Config Attributes (for any vm Name)

- Console cmd

```
device# show running-config iah
iah
    vm Ubuntu1
        description Ubuntu demo vm
        auto-boot true
        activate-network
        cpus 1
        ram 1024
        disk sda
            size 5000
            format qcow2
            bus sata
f        file disk://iah/disk1.qcow2
        !
        network nic2
            interface management 0
        !
    !
!
```

- 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:::trueno 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

- OS ONE Switching and Routing – When RME is configured, it utilizes the insight/ internal port, making it unavailable for other insight or internal purposes.

12. Configure the internal port for TPVM usage.

- Deploy a VM with the following specifications:
 - CPU: 3 vCPUs
 - RAM: 1 GB
 - Disk Space: 10 GB
 - Disk Image: QCOW2
 - Disk Bus Type: SATA
 - Auto-start VM on host reboot
- Complete the following network configuration:
 - Attach 1 management port
 - Attach 1 insight/internal port

```
device# conf t
device(config)# iah
device(config-iah)# vm ubuntu ← Name of the VM
device(config-iah-vm-ubuntu)# cpus 3
device(config-iah-vm-ubuntu)# ram 1024
device(config-iah-vm-ubuntu)# autostart
device(config-iah-vm-ubuntu)# description this is a ubuntu VM ← description
text.
device(config-iah-vm-ubuntu)# disk sda
device(config-iah-vm-disk-sda)# file disk://iah/ubuntu.qcow2 ← disk image file
should be downloaded to the /var/data/disk/iah/ before
device(config-iah-vm-disk-sda)# bus sata
device(config-iah-vm-disk-sda)# ex
device(config-iah-vm-ubuntu)# network nic1
```

```

device(config-iah-vm-network-nic1)# interface management 0
device(config-iah-vm-network-nic1)# ex
device(config-iah-vm-ubuntu)# network nic2
device(config-iah-vm-network-nic2)# interface internal 0
device(config-iah-vm-network-nic2)# ex
device(config-iah-vm-ubuntu)# deploy
device(config-iah-vm-ubuntu)# end
device# show iah vm ubuntu
device# iah vm console ubuntu
Connected to domain 'ubuntu'
Escape character is ^x (Ctrl + x) □ press control and X to exit VM console
.....
root@ubuntu-vm:~#
Closing console connection for VM: ubuntu
device#

```

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)# 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

```

14. 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

```

15. 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
```

16. 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::1/no 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
```

17. 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
```

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

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



Note

You can specify the config attribute only using the filename. Perform the file copy operation using the COPY exec command. In the RMA use case, you can make the CERT file available on the host, and use the filename to place it on the TPVM. You can use the filename to place it at TPVM.

- a. Copy SCP URL to a file at host.
- b. Copy from host to TPVM (can be done directly).

19. 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
peerUserName/* 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 OS ONE IAH.

OS ONE 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	<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.	<pre>rpc IahFileInfo(IahFileInfoReq) returns (IahFileInfoResp) {};</pre>

Console CLI	gRPC
Syntax iah vm TPVM snapshot create iah vm TPVM snapshot create with the fa iah vm TPVM snapshot revert iah vm TPVM snapshot delete	<pre>// VM snapshot RPC rpc VmSnapshot(VmSnapshotReq) returns (VmSnapshotResp) {}</pre>

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

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

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	<pre>rpc VmImport(VmImportReq) returns (VmImportResp) {}</pre>

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 OS ONE.

OS ONE 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: B
    disk: sda
        size: 5000 MB
        file: flash://iah/ubuntu-disk1.qcow2
        file format: QCOW2
        bus type: SATA
    network: 1
        interface: management 0
        connector-type: MANAGEMENT
        mac: e4:44:e5:08:ac:00
device#

device# show iah vm TPVM

Name: TPVM
    deploy-status: PERSONALITY_CONFIG_SUCCESS
    vm-status: RUNNING
    auto-boot: enabled
    activate-network: enabled
    TPVM:
        version: 4.7.9
        ssh-passwordless: enabled
device#

device# show iah vm TPVM details

show iah vm TPVM details
Name: TPVM
    deploy-status: PERSONALITY_CONFIG_SUCCESS
    vm-status: RUNNING
    last-error: -
    description: TPVM
    auto-boot: enabled
    activate-network: enabled
    cpus: 2

```

```

ram: 4096 MiB
disk: sda
  size: 10240 MB
  file: /mnt/onl/iah/images/TPVM/sda.qcow2
  file format: qcow2
  bus type: virtio
disk: sdb
  size: 40960 MB
  file: /mnt/onl/iah/images/TPVM/sdb.qcow2
  file format: qcow2
  bus type: virtio
disk: hda
  size: 1 MB
  file: /mnt/onl/iah/images/TPVM/hda.raw
  file format: raw
  bus type: ide
network: mgmt0
  interface: management 0
  connector-type: MANAGEMENT
  mac: 40:88:2f:fa:78:05
TPVM:
  hostname: TPVM-TD3-2
  user: extreme
  timezone: Asia/Kolkata
  version: 4.7.9
  ssh-passwordless: enabled
  ipaddress: 10.38.65.43/24
  gateway: 10.38.65.1
  ipv6address: -
  gatewayv6: -
  ntp:
    server: 10.64.12.81
device#

```

- 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 RASlog messages for OS ONE.

OS ONE 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 the following benefits:

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

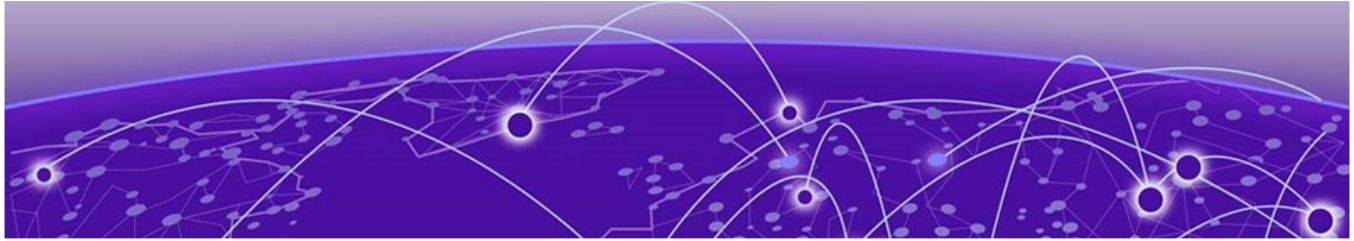
The bid (Internal Transaction Counter ID) helps track transactions and matches them with SupportSave data and timestamps.

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

Event Log Messages

Event ID	Event Log	Remarks
27001	Microservice initialization failed.	
27002	Error occurred during management service notification handling.	
27003	Error occurred in publishing state notifications.	
27004	Error during internal DB operation.	
27005	Error during communication with clients.	
27006	Error in starting GRPC server.	
27007	Error occurred in communicating with Tierra service.	
27008	Error occurred in communicating with HAL service.	
27009	Service shutting down.	
27010	Service is ready.	
27011	Invalid name.	name identifier validation failed.
27012	Invalid VM configuration.	Can not deploy VM with the given configurations.

Event ID	Event Log	Remarks
27013	No resources.	Not enough system resources.
27014	VM control command failed.	Failure in executing the control command.
27015	MAC address generation error.	Failure to generate uniq MAC address.
27016	XML parsing error.	Failure to parse libvirt XML file.
27017	Device attach error.	Failure to attach device to VM.
27018	Disk file error.	Disk image file not accessible or invalid format/size.
27019	No such VM.	Could not find VM with the given name.
27020	Virtual Machine <VM NAME> deployed.	VM deployment was successful.
27021	Virtual Machine <VM NAME> undeployed.	VM un deployment was successful.
27022	Virtual Machine <VM NAME> operationally UP	
27023	Virtual Machine <VM NAME> operationally DOWN	
27024	Virtual Machine <VM NAME> shutting down.	
27025	Virtual Machine <VM NAME> rebooting.	
27026	Virtual Machine <VM NAME> suspending.	
27027	Virtual Machine <VM NAME> resuming.	
27028	Virtual Machine <VM NAME> starting.	
27029	Virtual Machine <VM NAME> import file <FILE NAME>deleted successfully Virtual Machine <VM NAME> import file <FILE NAME> can be deleted	
27031	Libvirt error	Failed to communicate with libvirt.



Network Time Protocol and Clock Management

[Network Time Protocol](#) on page 119

[Configure NTP](#) on page 121

[Display NTP Server Status](#) on page 122

[NTP to Chrony Migration](#) on page 123

Network Time Protocol

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.

NTP Features

- Uses Coordinated Universal Time (UTC) to synchronize CPU clock time
- Avoids vulnerabilities in information exchange communication
- Can configure up to eight servers and eight peers to synchronize system clock
- Supports in-band configuration with VRF mapping to the NTP service

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

Use the Network Time Protocol server to obtain the correct time from an external time source and adjust the local time in each connected device. When you enable an NTP server functionality, the NTP server starts listening on the NTP port for client requests and responds with the reference time. You can configure up to eight server addresses 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

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**

See the RFC 5905.

4. Use the reference time from the system peer for system time synchronisation.
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.
- 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. Use the **enable** command to set the device to operate in server mode. Use the **no enable** command 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.

Configure 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
```

Display 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:

```
Clock is synchronized, stratum 4, reference clock is 0A400C51 (10.64.12.81)
precision is -24
reference time is Tue Aug 18 06:14:37 2026
clock offset is 0.000005851 seconds fast of NTP time, root delay is 0.026955949 seconds
root dispersion is 0.025363538 seconds
NTP client mode is enabled
```

NTP to Chrony Migration

Extreme OS ONE now uses **Chrony 4.5** instead of the legacy **ntpd** daemon for Network Time Protocol (NTP) time synchronization. This migration changes the backend implementation while preserving the existing CLI workflow and configuration model wherever possible.

Chrony consists of the following components:

- **chronyd** — the daemon that maintains system time.
- **chronyc** — the command-line utility used for monitoring and runtime control.

This migration also updates associated wrapper interactions, state collection, telemetry reporting, and CLI output formatting.

NTP vs Chrony

Feature	NTP	Chrony
Accuracy	Millisecond level	Microsecond level
Convergence Speed	Slow (30+ minutes)	Fast (5 minutes)
Resource Usage	Higher	Lower
Mobile Support	Poor	Excellent
Configuration	Complex	Simple

Limitations

- `reach` Field Not Available
 - The `reach` field (poll history in octal format) from `ntpd` is not exposed in the Chrony-based CLI.
- Filter Data Not Available
 - Detailed filter metrics including `filter delay`, `filter offset`, and `filter dispersion` are not available.
- Reference Timestamp Precision
 - Reference timestamps display second-level precision only; sub-second fractions are not included.

SNMP Notifications

Trap Names and OIDs	Varbinds	Description
ntpEntNotifModeChange 1.3.6.1.2.1.197.0.1	ntpEntStatusCurrentMode	
ntpEntStatusCurrentMode 1.3.6.1.2.1.197.1.2.1	NA	• <code>notRunning(1)</code> — NTP is not running. • <code>notSynchronized(2)</code> — NTP is not synchronized to any time source. • <code>noneConfigured(3)</code> — NTP is not synchronized and no reference is configured. • <code>syncToRemoteServer(6)</code> — NTP is synchronized to an upstream server. • <code>unknown(99)</code> — The NTP state is unknown.

`syncToRefclock(5)` is not supported.

Pre-Migration Checklist

- Back up current NTP configuration files
- Document current NTP settings and server list
- Schedule migration during approved maintenance window
- Notify stakeholders of planned maintenance
- Test migration process in non-production environment first

Before validating NTP operation after migration, note the following:

- Existing NTP commands and workflows continue to work.
- Backend time synchronization is now handled by Chrony instead of `ntpd`
- Enable and disable workflows are preserved.
- Server and peer configuration behavior remains consistent with earlier workflows.
- VRF-based deployments continue to use the existing mechanism and do not require workflow changes.
- Command output may differ from earlier releases because the system now reports Chrony-derived runtime data instead of `ntpd` data.

CLI Output: NTP vs Chrony

Show NTP Status

The **show ntp status** command confirms whether the system clock is synchronized and identifies the active reference source. Use this command to review configured NTP sources and determine which source is currently selected.

Old Output (NTP)

```
Clock is synchronized, stratum 3, reference clock is 10.64.12.81,
precision is -23,
reference time is edad680a.74edd509 Tue, May 12 2026 8:49:14.456,
clock offset is +0.000000, root delay is 292.665,
root dispersion is 1118.455, peer dispersion is 48670,
NTP client mode is enabled
```

New Output (Chrony)

```
Clock is synchronized, stratum 3, reference clock is 0A400C51 (10.64.12.81)
precision is -24
reference time is Tue May 12 17:18:57 2026
clock offset is 0.000000000 seconds slow of NTP time, root delay is 0.292580664 seconds
root dispersion is 0.299098760 seconds
NTP client mode is enabled
```

Show NTP Association

The **show ntp association** command displays the configured NTP sources along with stratum, polling information, reachability, last receive time, and the most recent time sample.. Use this command when you want a quick operational summary of source selection, reachability, and synchronization quality.

Old Output (NTP)

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*10.64.12.81	23.93.251.54	2	u	1	64	1	216.051	+0.405	0.226

* synced, # selected, + candidate, - outlayer, x falseticker, ~ configured

New Output (Chrony)

MS Name/IP address	Stratum	Poll	Reach	LastRx	Last sample
^* 10.64.12.81	2	6	17	17	+14us[+16us] +/- 441ms

Mode: ^ server, = peer, # local clock | State: * current best, + combined, - not combined, x may be in error, ~ too variable, ? unusable

Show NTP Association Detail

The **show ntp association detail** command. Use this command when you need a detailed diagnostic view of NTP behavior for validation or troubleshooting.

The command provides detailed source information including local and remote addresses, leap status, protocol version, mode, stratum, precision, root delay, root dispersion, reference time, offset, peer delay, peer dispersion, response time, timestamping, and transmit/receive counters.

Old Output (NTP)

```

10.64.12.81:123 configured, best candidate, stratum 2,
ref ID 23.93.251.54, time edad6407.80177e95 Tue, May 12 2026 8:32:07.500,
our mode client, peer mode server, our poll intvl 6, peer poll intvl 6,
root delay 76.614 msec, root disp 180.573, reach 001, jitter 0.185,
delay 216.083 msec, offset +0.422 msec, dispersion 187.530,
precision is 2**-24, version 4
filter delay = ...
filter offset = ...
filter disp = ...

```

New Output (Chrony)

```

10.64.12.81:123 configured, best candidate
Remote address : 10.64.12.81 (0A400C51)
Remote port    : 123
Local address  : 10.37.37.5 (0A252505)
Leap status    : Normal
Version        : 4
Mode           : Server
Stratum        : 2
Poll interval  : 6 (64 seconds)
Precision      : -24 (0.000000060 seconds)
Root delay     : 0.077820 seconds
Root dispersion : 0.294525 seconds
Reference ID    : 175DFB36 ()
Reference time  : Tue May 12 14:58:38 2026
Offset         : -0.000016378 seconds
Peer delay     : 0.214760855 seconds
Peer dispersion : 0.000429798 seconds
Response time  : 0.000081662 seconds
Jitter asymmetry: +0.00
NTP tests      : 111 111 1111
Interleaved    : No
Authenticated  : No
TX timestamping : Kernel
RX timestamping : Kernel
Total TX       : 4
Total RX       : 4
Total valid RX : 4
Total good RX  : 4

```

Show Running-State System NTP

The **show running-state system ntp** command reviews the NTP operational state in the running-state hierarchy. Use this command when you want operational state data aligned to the current telemetry and YANG-based reporting model.

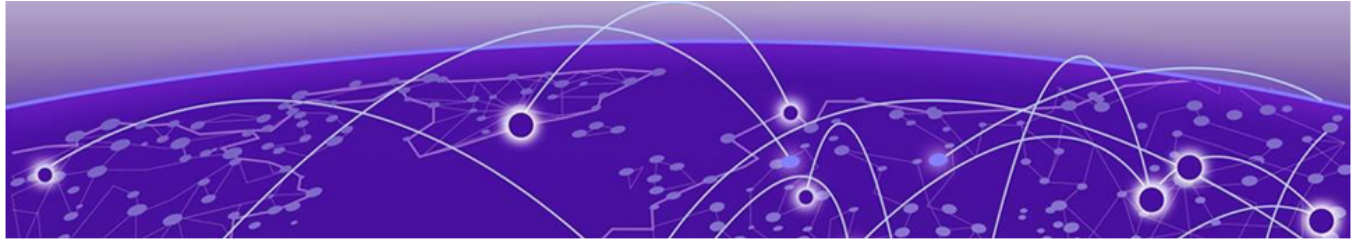
Old Output (NTP)

```

system
 ntp
   server 10.64.12.81
     stratum 2
     offset 0.422ms
     poll-interval 64s
     root-delay 216.083ms
   !
   enable
   active-server 10.64.12.81
   active-server-offset 0.422ms
 !
!
```

New Output (Chrony)

```
system
 ntp
  server 10.64.12.81
  stratum 2
  offset 14000ns
  poll-interval 64s
  root-delay 77820000ns
  root-dispersion 294525000ns
  port 123
  version 4
!
enable
active-server 10.64.12.81
active-server-offset 14000ns
!
!
```



SNMP

[How SNMP Monitors Network Devices](#) on page 128

[Configure SNMPv2](#) on page 138

[Configure SNMPv3](#) on page 139

[Supported MIBs](#) on page 140

[SNMP Monitoring for Fan and Power Supply Status](#) on page 140

[SNMP Monitoring for Optical Transceiver](#) on page 147

How SNMP Monitors Network Devices

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

SNMP enables you 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 you to track specific data values.

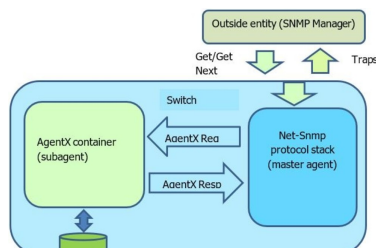
The SNMP server on the Extreme Networks device supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

You 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. An open-source 'net-snmp' protocol stack provides the SNMP support in Extreme OS ONE.

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.

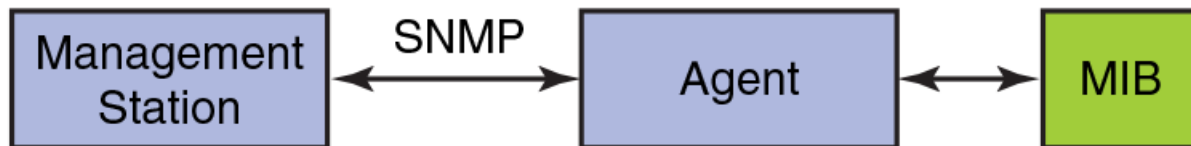


Figure 1: SNMP Structure

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, you can filter for traps from an 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

SNMP Engine ID, System Description, Groups, and Views

Release **22.2.3.0** adds support for three SNMP enhancements in Extreme OS ONE: a user-configurable **SNMP engine ID**, a user-configurable **SNMP system description (sysDescr)**, and **SNMP groups and views** for access control. These capabilities are available through both **CLI** and **gNMI**.

Supported Capabilities

The release supports the following SNMP configuration areas:

- **SNMP engine ID** configuration through CLI and gNMI.
- **SNMP sysDescr** configuration through CLI and gNMI.
- **SNMP views and groups**, including association of views with groups and association of groups with SNMPv1/v2c communities and SNMPv3 users.

SNMP Engine ID

The system now supports a **user-configurable SNMP engine ID**. In earlier releases, the engine ID was generated internally and could not be changed. Starting with release 22.2.3.0, you can configure the engine ID manually.

Engine ID behavior

- The configured engine ID must be **5 to 32 bytes** in length.
- Enter the value in **hexadecimal format**, with bytes separated by colons.
- The internally generated default engine ID is **11 bytes** long.
- When you update the engine ID, the **SNMP server restarts internally** for the change to take effect.
- A user-configured engine ID is retained across **reboots** and **firmware upgrades**. It is removed after **fullinstall**, **factory reset**, and **ONIE install** operations.
- The **default engine ID** remains visible in the running-state output.

Example engine ID format

```
80:0:6:34:b2:6e:f4:0:10:76:18:9f:95
```

SNMP System Description (sysDescr)

The system now supports a **user-configurable SNMP system description**. This allows you customize the value returned for the **sysDescr** object in **SNMPv2-MIB**.

sysDescr behavior

- The configured sysDescr can contain up to **255 characters**.
- By default, the sysDescr value matches the system description shown in the **show sysinfo chassis** output.
- A user-configured SNMP sysDescr affects the **sysDescr MIB object** only. It does **not** change the chassis system description displayed by **show sysinfo chassis**.
- A user-configured sysDescr is retained across **reboots** and **firmware upgrades**. It is removed after **fullinstall**, **factory reset**, and **ONIE install** operations.
- The **default sysDescr** remains visible in the running-state output.

SNMP Groups and Views

Release 22.2.3.0 adds support for the **View-based Access Control Model (VACM)** for SNMP. VACM enables administrators to control access to MIB objects by defining **views**, organizing SNMP entities into **groups**, and associating users or communities with those groups.

In this access model:

- **Views** define which parts of the MIB tree are accessible.
- **Groups** define access policies and role-based behavior.
- **View-group associations** apply **read** and **notify** permissions to a group.
- **Community and user bindings** map SNMPv1/v2c communities and SNMPv3 users to a specific group.

This model provides granular access control and helps ensure that each SNMP entity can access only the MIB objects that are allowed for its role.

Scale limits

- You can configure a maximum of **10 groups**.
- You can configure a maximum of **10 views**.
- Each view can contain up to **10 OIDs total**, including both **included** and **excluded** OIDs.
- The same OID cannot be both **included** and **excluded** in the same view. If that occurs, the system returns an error.

Unsupported Capabilities and Limitations

The following capabilities are not supported in this release:

- **Write views** are not supported. Only **read** and **notify** views are supported for groups.
- **VACM MIB** support is not included.
- **Context-based access control** is not supported.

Configuration Components

The implementation introduces CLI and data model support for the following configuration objects:

- **engine-id** for SNMP engine ID configuration.
- **sys-descr** for SNMP system description configuration.
- **view**, along with **include** and **exclude**, to define accessible OID subtrees.
- **group**, **version**, **read-view**, **notify-view**, and **auth-level** to define group-based access policy.
- **groupname** under communities and users to associate them with a configured group.

The YANG data model includes support for **engine-id**, **sys-descr**, **views**, **groups**, **community-to-group mappings**, and **user-to-group mappings** under the SNMP server hierarchy.

Configure SNMP Access Control

Keep the following points in mind when you deploy the SNMP enhancements (SNMP Engine ID, System Description, Groups, and Views):

- Changing the engine ID causes an internal SNMP server restart. Plan the update accordingly.
- A custom sysDescr changes the value exposed through SNMP, but it does not change chassis information shown in local system commands.
- Group and view design directly controls SNMP access scope. Define views carefully to prevent unintended access to MIB objects.
- Because write views are not supported, this release is limited to **read** and **notify** access control for VACM-based group configuration.

Follow this high-level workflow procedure to configure SNMP access control.

1. Configure the SNMP engine ID, if you need a value other than the system-generated default.

Configures the new SNMP engine ID and restarts the SNMP server for it to take effect.

```
device (config-system-snmp-server)# engine-id 1a:2b:3c:ab:a1:ba:1a:9a:5a:1a:2a:2b
```

2. Configure the SNMP sysDescr, if you want to customize the sysDescr object returned to SNMP managers.

```
device (config-system-snmp-server)# sys-descr Test System Description
```

3. Create one or more views and define which OIDs are included or excluded.

```
device (config-system-snmp-server)# view view1
```

4. Create one or more groups and assign read and notify views to each group.

```
device (config-system-snmp-server)# group group1
```

5. Associate SNMPv1/v2c communities or SNMPv3 users with the appropriate group.

```
device (config-system-snmp-server)# community c1
device (config-system-snmp-server-community-c1)# group groupro

device (config-system-snmp-server)# user u1 auth noauth
device (config-system-snmp-server-user-u1)# group groupro
```

Example Configuration

The following example shows a running configuration with a custom sysDescr, a configured engine ID, a view with included and excluded OIDs, groups for SNMPv1 and SNMPv3 access, and community/user-to-group bindings.

The example demonstrates how views control accessible OIDs, how groups define access behavior, and how communities or users inherit access through group association.

```
# show running-config system snmp-server
system
snmp-server DEFAULT
sys-descr x86-64 Test System Description
engine-id 1a:2b:3c:ab:a1:ba:1a:9a:5a:1a:2a:2b
view view1
include 1.3.6.1.2.1.2
include 1.3.6.1.2.1
exclude 1.3.6.1.2.1.2.6
exclude 1.3.6.1.2.1.2.8
!
group group1
version v1
read-view view1
notify-view view1
!
!
group group3
version v3
read-view view1
auth-level auth
!
!
community extremenet
groupname group1
```

```
!
user user1 auth md5 auth-password-hashed 0x285495b04b8895f5462e7722f4d92481 priv
nopriv
  groupname group3
!
vrf mgmt-vrf
  host 10.37.32.34 extremenet
  version 2c
  udp-port 162
!
  enable
!
!
```

Upgrade and Downgrade Behavior

Upgrade from Earlier Releases

When you upgrade from a lower release:

- The system continues to use the **default engine ID** until you explicitly configure a custom engine ID.
- In **22.2.3.0**, the default engine ID and sysDescr are updated in the state database under the SNMP server regardless of whether SNMP is enabled or disabled.
- If community or user configurations already exist, the system automatically adds **default groups** and **default views** so that those communities and users continue to function after the upgrade.
- Communities are mapped to a **default_group** with **v1** and **v2c** versions configured and a **default_view** applied as both the read view and notify view. That default view includes the root OID 1.
- Users are mapped to default groups based on their configured authentication level. Separate default groups are created for **noauth**, **auth**, and **priv** user types as needed.

Downgrade to Earlier Releases

When you downgrade to a lower release:

- Any **custom engine ID** configuration is lost, and the system reverts to the default engine ID.
- If an SNMP community or user is associated with a group, you must remove those group associations before the downgrade because earlier releases do not support them.

Event Log Messages

The SNMP agent introduces the following event log messages:

- **36001** — *SNMP server started successfully.*

This indicates that the SNMP server started successfully. No action is required.

- **36002** — *SNMP server stopped successfully.*

This indicates that the SNMP server stopped successfully. SNMP service is not available while it is stopped.

- **36003** — *Failed to start SNMP server.*

This indicates that the SNMP server failed to start. The recommended action is to restart the server.

- **36004** — *SNMP coldstart trap has been sent.*

This indicates that the coldStart trap was sent successfully. No action is required.

- **36005** — *SNMP Engine ID updated. SNMP server will be restarted for new engine ID to take effect.*

This indicates that a new engine ID has been stored and that the SNMP server will restart internally.

SNMP Informs

SNMP Informs support extends SNMP notification handling by providing acknowledged delivery for SNMPv2c and SNMPv3 hosts. Unlike SNMP Traps, which are sent without confirmation, SNMP Informs require a response from the receiving SNMP manager. If the response is not received within the timeout period, the system retransmits the Inform request to improve notification reliability.

This capability also introduces per-host selection of notification types, support for a remote engine ID for SNMPv3 Inform targets, and an explicit enable step to commit host configuration only after all required parameters are set.

The following new capabilities have been added:

Feature	Description
SNMP Informs	Reliable notification delivery with acknowledgment for SNMPv2c and SNMPv3 hosts (Inform notifications). Unlike traps, the system automatically retransmits if no acknowledgement is received.
Remote Engine ID	Configurable unique identifier for the remote SNMP manager. Allow configuration of a remote engine ID for SNMPv3 Inform destinations. Enables proper authentication for SNMPv3 Informs.
Notification Type Selection	Allows notification type selection on a per-host basis, with support for both Traps and Informs.
Enable Control	Explicitly activate host configurations after setup, preventing incomplete host configurations from being committed until the host is explicitly enabled.

Understanding Traps vs. Informs

SNMP provides two methods for sending network notifications. Understanding the differences helps you choose the right approach for your monitoring needs:

Characteristic	Traps	Informs
Delivery Confirmation	None (fire-and-forget)	Required (acknowledged)
Reliability	Best effort only	Guaranteed with automatic retry
Protocol Versions	SNMPv1, SNMPv2c, SNMPv3	SNMPv2c, SNMPv3 only

Characteristic	Traps	Informs
Resource Usage	Low	Moderate (with retransmission)
Best Use Case	High-volume events, non-critical alerts	Critical events requiring confirmation

Configuration Support by Protocol Version

Different SNMP protocol versions have varying capabilities. This matrix shows which features are supported for each:

SNMP Version	Traps Support	Informs Support	Engine ID Support
SNMPv1	Yes (default)	Not supported	Not applicable
SNMPv2c	Yes (default)	Supported	Not applicable (ignored)
SNMPv3	Yes (default)	Supported	Required for Informs

Operational Considerations

Upgrade Behavior

- Existing host configurations remain functional with the default notify-type of TRAPS.
- Existing host configurations are automatically enabled (that is, committed to `snmpd.conf`) as part of the migration.

Downgrade Behavior

- When downgrading to a lower firmware version, any hosts configured with notify-type INFORMS must be removed before the downgrade can proceed.
- The `enable` attribute is not supported in lower versions; hosts are committed by default, and this attribute is discarded during the downgrade.

Performance Considerations

Informs require acknowledgment and may be retransmitted, so they consume more resources than Traps. The documented impact is minimal for typical notification volumes.

Security Considerations

SNMPv3 Informs use the existing User-based Security Model (USM) for authentication and encryption. No additional security risks are introduced beyond the existing SNMPv3 security model.

Licensing

No additional license is required. This functionality is available with the base license.

Configure SNMP Informs

Follow this procedure to configure the SNMP host for Informs.

- Create the SNMP host entry with the required host address and community or user information.

2. Set the host version.
3. Configure the UDP port if required.
4. Set `notify-type` to `INFORMS` for SNMPv2c or SNMPv3 hosts.

```
device(config-system-snmp-server-vrf-mgmt-vrf-host-1.2.3.4-****)# notify-type informs
```

The `notify-type informs` is applicable only for SNMPv2c or SNMPv3 hosts. The system returns an error if you try to configure informs for an SNMPv1 host. SNMPv1 does not support informs. By default, the host will be sent traps.

5. For SNMPv3 hosts, configure the remote `engine-id` if the manager `engine-id` is known.

```
device(config-system-snmp-server-vrf-mgmt-vrf-host-1.2.3.4-****)# engine-id
80:00:01:02:03:04:05:06
```

Remote `engine-id` is applicable only for SNMPv3 informs. If this value is set for any other host or `notify-type` combination, it will be accepted but ignored.

If `engine-id` is not provided for SNMPv3 informs, `net-snmp` learns `engine-id` through the discovery mechanism.

6. Enable the host configuration to commit it to the SNMP agent configuration file.

```
device(config-system-snmp-server-vrf-mgmt-vrf-host-1.2.3.4-****)# enable
```

This configuration allows you to enable the given host. As a best practice, use this command after configuring all host parameters. When the host is already enabled, modification of any host parameter is allowed.

Configure 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-mgmt-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)# enable
```

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
    notify-type informs
    engine-id 80:00:01:02:03:04:05:06
    enable
  !
!
!
!
device#
```

Configure 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 DEFAULT
```

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)# enable
```

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
    notify-type informs
    engine-id 80:00:01:02:03:04:05:06
    enable
  !
!
!
!
```

Supported MIBs

Extreme OS ONE supports standard and enterprise MIBs and traps. For a complete list and details, see the *Extreme OS ONE Switching and Routing SNMP MIB Reference Guide*.

SNMP Monitoring for Fan and Power Supply Status

Extreme OS ONE Switching and Routing support **SNMP monitoring and notification for field-replaceable units (FRUs)** such as **fan trays** and **power supply units (PSUs)**. This

support allows you to perform SNMP walk operations to view FRU status and receive SNMP trap notifications when a fan or power supply fails or recovers.

This functionality uses the **extremePlatformMIB** under the Extreme enterprise OID tree and sends the **fruStatusChanged** notification to configured SNMP trap receivers when the operational state of a monitored FRU changes.

This implementation provides the following capabilities:

- SNMP walk support for **fan** and **power supply** status.
- SNMP trap notifications for **fan failure** and **fan recovery** events.
- SNMP trap notifications for **power supply failure** and **power supply recovery** events.
- A new **extremePlatformMIB** definition rooted under **extremeAgent.65**.
- Trap varbinds that include **entPhysicalName**, **fruStatus**, **fruClass**, **fruObjectNum**, and **sysName**



Note

- SNMP SET operations on FRU table objects (all objects are read-only) are not supported.
- Traps are generated only on state transitions (failure/recovery) and not periodically while the FRU remains in a failed state..

How FRU Monitoring Works

The platform monitors fans and power supplies as FRUs and reports state changes through SNMP. When a fan or power supply transitions to a failure or recovery state, the switch generates an SNMP trap and sends it to all configured trap receivers.

Trap generation occurs only when the FRU **changes state**. The switch does not send repeated notifications while the FRU remains in the same failed state. This behavior reduces unnecessary overhead and avoids continuous trap generation.

SNMP Notification Behavior

The switch sends the **fruStatusChanged** notification when the status of a monitored FRU changes. The notification applies to both fans and power supplies.

Fan notification triggers

A trap is generated in the following cases:

- **Fan failure**: when the fan transitions to **faulty** status. In the trap, **fruStatus=5** and **fruClass=9**.
- **Fan recovery**: when the fan transitions from **faulty** to **on**. In the trap, **fruStatus=3** and **fruClass=9**.

Power supply notification triggers

A trap is generated in the following cases:

- **Power supply failure:** when the PSU transitions to **faulty** status. In the trap, **fruStatus=5** and **fruClass=8**.
- **Power supply recovery:** when the PSU transitions to **on** from a failed or present-but-not-powered state. In the trap, **fruStatus=3** and **fruClass=8**.

MIB Structure and Location

File Location

- `snmp_agent/mibs/enterprise/extremePlatformMIB.my`

OID Tree Location

- **Root:** `extremeAgent.65` (1.3.6.1.4.1.1916.1.65)
- **Enterprise OID:** 1.3.6.1.4.1.1916 (Extreme Networks)
- **Agent Subtree:** 1.3.6.1.4.1.1916.1.65 (extremePlatformMIB)

FRU Status Changed Notification

The `fruStatusChanged` trap is generated whenever the status of any FRU object (fan or power supply) changes. This notification enables real-time alerting on component status changes in network management systems.

Table 5: Notification Definition

Trap Name	OID	Varbinds
<code>fruStatusChanged</code>	1.3.6.1.4.1.1916.1.65.2.0.1	<code>entPhysicalName</code> , <code>fruStatus</code> , <code>fruClass</code> , <code>fruObjectNum</code> , <code>sysName</code>



Note

The `fruStatusChanged` notification carries comprehensive information about the affected FRU, allowing management systems to automatically identify and respond to component changes.

Trap Varbind Specifications

Each fruStatusChanged trap contains the following varbinds (variable bindings) that provide detailed information about the status change:

OID	Object	Type	Description
1.3.6.1.2.1.47.1.1.1.1.7	entPhysicalName	Octet String	The name of the physical entity. For fans: "Fan N". For power supplies: "Power Supply N". (1-based index)
1.3.6.1.4.1.1916.1.65.1.5.1.2	fruStatus	Integer	The current status of the FRU: 1 (other), 2 (unknown), 3 (on), 4 (off), 5 (faulty).
1.3.6.1.4.1.1916.1.65.1.5.1.1	fruClass	Integer	The type of FRU: 8 (powerSupply), 9 (fan). See FruClass textual convention below.
1.3.6.1.4.1.1916.1.65.1.5.1.3	fruObjectNum	Integer32	The unit number of the FRU (0-based index).
1.3.6.1.2.1.1.5	sysName	Octet String	The administratively-assigned name for this managed node (hostname).

FRU Classification Reference

The FruClass textual convention defines the type of FRU object. Use these values when interpreting fruClass varbind data:

Value	Name	Description
1	other	Unspecified component
2	unknown	Unknown component type
3	chassis	Chassis or enclosure
4	cp	Control processor module
5	other-CP	Alternative control processor
6	switchblade	Switch blade module
7	wwn	Wide area network module
8	powerSupply	Power supply unit
9	fan	Cooling fan module
10	coreblade	Core processing blade
11	applicationblade	Application blade module



Note

For FRU monitoring, focus on fruClass values 8 (powerSupply) and 9 (fan). Other values may represent chassis components in modular systems.

FRU Status Values

The fruStatus object indicates the operational state of an FRU. The following table defines all possible values:

Value	Status	Meaning
1	other	Unspecified state (verify manually)
2	unknown	Status cannot be determined (possible communication failure)
3	on	Component is operational (normal state)
4	off	Component is powered down or inactive (normal for redundancy)
5	faulty	Component has failed (requires immediate action)

Recommended Alert Thresholds

- **unknown, faulty:** CRITICAL alert—immediate investigation required
- **other:** WARNING—unexpected state; verify device condition
- **off:** Normal for redundant components if N+1 protection active
- **on:** Normal operating state

Event Logging

The switch also records FRU state changes in the existing RAS logs. Fan and PSU enable, failure, and extraction events are logged by the chassis management service. Additional debug-level messages can be generated for trap encode errors, publish errors, and successful trap transmission.

Examples of existing event categories include the following:

Log ID	Level	Service	Description
5300	Info	chassis-mgr	Fan enabled (operational) with status, RPM, RPM%
5301	Warning	chassis-mgr	Fan failed
5302	Info	chassis-mgr	Fan departed (extracted)
5400	Info	chassis-mgr	PSU enabled (operational) / Invalid PSU ID
5401	Warning	chassis-mgr	PSU failed
5402	Info	chassis-mgr	PSU departed (extracted)

Considerations and Limitations

- The switch generates notifications only on **state transitions** and does not resend notifications periodically while the same failed state remains active.
- Trap generation is **asynchronous** and does not block chassis monitoring. If the message bus publish fails, the error is logged but does not affect the FRU state management.

- No new CLI commands are introduced for this functionality. Existing `show sysinfo` commands are used for operational verification.
- No license is required to use this functionality.
- No XML impact or UPM interaction is introduced.
- The functionality does not collect, store, or transmit personal data.

Security Considerations

SNMP trap delivery uses the switch's existing SNMP setup - that means it'll rely on your current SNMPv2c community strings or SNMPv3 credentials, whatever you've already configured.

Monitor Fan and Power Supply Status Using SNMP

Ensure that the following conditions are met:

- SNMP is enabled on the device.
- One or more **SNMP trap receivers** are configured on the switch.
- Your management system can access the **extremePlatformMIB** and process Extreme enterprise OIDs.
- Appropriate SNMP security settings are already configured on the switch, such as **SNMPv2c community strings** or **SNMPv3 credentials**.



Note

This functionality does not introduce new security mechanisms. Trap delivery relies on the existing SNMP configuration on the switch.

Follow this procedure to monitor fan and power supply status by using SNMP.

1. Verify fan failure and recovery.
 - a. Configure an SNMP trap receiver on the switch.
 - b. Remove a fan tray or simulate a fan failure.
 - c. Run `show sysinfo fan` and confirm that the failed fan is missing from the output or no longer reported as operational.

```
show sysinfo fan
```

The following is an example output:

```
device# show sysinfo fan
```

Fan Information

Id	Status	RPM	Percentage	SpeedLevel	Direction

1	Up	7400	30	LOW	FAN_DIR_F2B
2	Up	7400	30	LOW	FAN_DIR_F2B
3	Up	7400	30	LOW	FAN_DIR_F2B
4	Up	7400	30	LOW	FAN_DIR_F2B
5	Up	7400	30	LOW	FAN_DIR_F2B
6	Up	7600	30	LOW	FAN_DIR_F2B
7	Up	7400	30	LOW	FAN_DIR_F2B

If a fan is not present or has failed, the output can show fewer rows than expected. A missing fan entry indicates a failure condition that can trigger an SNMP trap.

- d. Confirm that the trap receiver receives `fruStatusChanged` with `fruClass=9`, `fruStatus=5`, and `entPhysicalName="Fan N"`.
 - e. Reinsert the fan tray and confirm that the fan returns to Up status.
 - f. Confirm that a recovery trap is received with `fruStatus=3`.
2. Verify power supply failure and recovery.
 - a. Configure an SNMP trap receiver on the switch.
 - b. Unplug or power down a PSU, or insert a PSU that is present but not powered up.
 - c. Run `show sysinfo power-supply` and verify the PSU status on the switch.

```
show sysinfo power-supply
```

The following is an example output where PSU 2 is present but not powered up. The platform monitors this status and can send notifications when the PSU changes to a failed or recovered state.:

```
device# show sysinfo power-supply
PSU Information
Id      Status      Type   C[in]  C[out] P[in]  P[out] V[in]  V[out]
-----
1       Up             AC     1.200  18     280    220    224    11
2       Present        AC     0       0       0       0       0       0
```

- d. Confirm that the trap receiver receives `fruStatusChanged` with `fruClass=8`, `fruStatus=5`, and `entPhysicalName="Power Supply N"`.
 - e. Restore power to the PSU and confirm that the device reports the PSU as operational.
 - f. Confirm that a recovery trap is received with `fruStatus=3`.
3. View power supply status.

Run the following command to display power supply status on the switch:

```
show sysinfo power-supply
```

The following is an example output where PSU 2 is present but not powered up. The platform monitors this status and can send notifications when the PSU changes to a failed or recovered state.:

```
device# show sysinfo power-supply
PSU Information
Id      Status      Type   C[in]  C[out] P[in]  P[out] V[in]  V[out]
-----
1       Up             AC     1.200  18     280    220    224    11
2       Present        AC     0       0       0       0       0       0
```

4. Verify trap delivery.

Verify trap delivery by using a trap receiver such as `snmptrapd` on a remote host. Capture the trap and confirm that it includes the expected notification OID and varbinds.

Verify that the trap contains the following elements:

- Notification OID: **1.3.6.1.4.1.1916.1.65.2.0.1**
- **entPhysicalName**
- **fruStatus**
- **fruClass**

- **fruObjectNum**
- **sysName**

SNMP Monitoring for Optical Transceiver

Learn about how to monitor optical transceivers using SNMP on Extreme Networks platforms running OS ONE Switching and Routing. It covers polling Digital Diagnostics Monitoring (DDM) data, interpreting SNMP MIB objects, and handling trap notifications for threshold events and optic qualification status.

Optical transceivers (SFP, SFP28, QSFP, QSFP28, QSFP-DD) expose real-time metrics such as temperature, voltage, transmit (Tx) power, receive (Rx) power, and bias current. These values can be accessed using SNMP and used by a Network Management System (NMS) for monitoring and alerting.

The SNMP optical monitoring implementation provides the following capabilities:

- Poll DDM sensor values per port and per lane using SNMP GET/GETNEXT
- Retrieve transceiver inventory details such as vendor, part number, and serial number
- Receive SNMP trap notifications on threshold crossings
- Detect and report unqualified optical transceivers

Supported metrics include:

- Temperature
- Supply voltage
- Tx power and Rx power
- Tx bias current

Event Log Messages

Level	Service	Description
error	fwd-hal	SFP sensor exceeded HIGH ALARM threshold
warn	fwd-hal	SFP sensor exceeded HIGH WARNING threshold
warn	fwd-hal	SFP sensor dropped below LOW WARNING threshold
error	fwd-hal	SFP sensor dropped below LOW ALARM threshold
info	fwd-hal	SFP sensor returned to normal
warn	fwd-hal	CheckSfp: Failed to fetch sensor thresholds from EEPROM
info	fwd-hal	CheckSfp: Unqualified SFP transceiver is detected
debug	fwd-hal	publishTrap: optical trap publisher not initialised, dropping trap
error	fwd-hal	publishTrap: encode failed / publish failed

Threshold Transitions

Events are generated when a metric transitions between threshold states. Each transition is encoded as an integer (1–20) representing the "from → to" state.

Threshold Transition Table (from → to)

	→Normal	→HiWarn	→HiAlarm	→LoWarn	→LoAlarm
Normal	—	3	1	4	2
HiWarn	13	—	14	16	15
HiAlarm	5	7	—	8	6
LoWarn	17	20	18	—	19
LoAlarm	9	11	10	12	—

Sample RAS Logs

Temperature high alarm:

```
2026-04-21 06:04:04.0779 fwd-hal[75]: Level:error LogID:0 Topic:1 Msg:SFP sensor exceeded
HIGH ALARM threshold sfp:ethernet 0/17 threshold:73 from:normal to:highAlarm transition:1
transition-name:normalToHighAlarm channel:1 metric:temperature value:75
```

Rx power low warning:

```
2026-04-21 06:04:04.0779
fwd-hal[75]: Level:error LogID:0 Topic:1 Msg:SFP sensor dropped below LOW ALARM
threshold sfp:ethernet 0/17 threshold:-13.50655141287858 from:normal
to:lowAlarm transition:2 transition-name:normalToLowAlarm channel:4
metric:input-power value:-14.672456210075023
```

SNMP Trap Notification Types

The system generates traps when threshold transitions occur or when an unqualified optic is detected.

Trap Name	OID	Trigger Condition	Varbinds
extremeIfTempStateChange	.1.3.6.1.4.1.1916.1.65.0.1	Temperature level changes (ch0 only)	extremeOptMonTemperature, extremeOptMonTemperatureEvent
extremeIfVoltStateChange	.1.3.6.1.4.1.1916.1.65.0.2	Voltage level changes (ch0 only)	extremeOptMonVoltageValue, extremeOptMonVoltageEvent
extremeIfTxPwrStateChange	.1.3.6.1.4.1.1916.1.65.0.3	Tx power level changes (per channel)	extremeOptMonTxPowerStatus, extremeOptMonTxPowerVal, extremeOptMonTxPower, extremeOptMonTxPowerEvent

Trap Name	OID	Trigger Condition	Varbinds
extremeIfRxPwrStateChange	.1.3.6.1.4.1.1916.1.65.0.4	Rx power level changes (per channel)	extremeOptMonRxPowerStatus, extremeOptMonRxPower, extremeOptMonRxPowerVal, extremeOptMonRxPowerEvent
extremeIfTxBiasStateChange	.1.3.6.1.4.1.1916.1.65.0.5	Tx bias current level changes (per channel)	extremeOptMonTxBiasCurrent, extremeOptMonTxBiasEvent
extremeIfOpticUnQualified	.1.3.6.1.4.1.1916.1.65.0.6	Unqualified optic detected on insertion	ifIndex

Monitor Optical Transceivers Using SNMP

Learn about the four comprehensive methods to test SNMP MIB configuration and transceiver monitoring on Extreme switches. These verification procedures ensure proper functionality of optical transceiver monitoring, trap generation, and Digital Diagnostic Monitoring (DDM) capabilities.

The following table summarizes the four test methods, their primary objectives, and expected outcomes:

Method	Objective	Requirements	Key Verification
SNMP Query	Verify MIB accessibility and port enumeration	Management station with SNMP query tools	MIB tables populated for all ports
DDM Override	Validate threshold trap generation	Transceiver, trap receiver, override file capability	Correct state transition traps received
Unqualified Optic	Verify qualification detection	Non-approved transceiver, CLI access	Unqualified trap and CLI confirmation
PDU Analysis	Verify message structure and content	SNMP trap daemon, event triggers	OID correctness, complete varbind set

Follow one of the methods to monitor optical transceivers using SNMP.

1. **Method 1:** SNMP Query Verification

This method uses SNMP GET and WALK operations to verify that the management information base is properly accessible and returns data for all optical ports.

Run the following command from your network management station to enumerate the optical transceiver monitoring table:

```
snmpwalk -v2c -c public <switch-ip> 1.3.6.1.4.1.1916.1.65
```

Where <switch-ip> represents the IP address of the target switch.

- Confirm that `extremeOptMonInfoTable` contains entries for all optical ports on the switch. Each row corresponds to a port configured or equipped with an optical transceiver.
- For multi-lane transceivers (QSFP or QSFP-DD), verify that `extremeOptMonLaneTable` contains multiple rows per physical port. Each row represents an individual lane within the multi-lane module.
- Confirm that `extremeIfMediaInfoTable` displays vendor identification, part number, serial number, and firmware revision for each port. This information is essential for inventory management and compatibility verification.

2. **Method 2:** Threshold Trap Verification via DDM Override.

This method injects synthetic sensor readings to validate trap generation without requiring optical transceivers with marginal power levels. The DDM override mechanism allows you to inject artificial readings for temperature, voltage, power, and bias current.

- a. Configure the trap receiver on the switch to deliver SNMP notifications to your network management system:

```
snmp-server host <nm-ip> traps version 2c public
```

Where <nm-ip> is the IP address of your network management system receiving traps.

- b. Insert an optical transceiver module into the port to be tested. Verify that the module is recognized by the switch.
- c. Create a DDM override file at `/tmp/sfp_ddm_override/ethernet017` (replace `ethernet017` with your target port designation). Populate the file with sensor values that exceed configured alarm thresholds.

Example values:

- `temp=100`
 - `voltage=10.0`
 - `rxpower=-35`
 - `txbias=80`
- d. Allow approximately one monitoring polling cycle (typically 1 second) for the switch to read the override values and detect the threshold violation.

- e. Monitor the trap receiver for an `extremeIfTempStateChange` notification. Verify that the received trap includes the `varbind extremeOptMonTemperatureEvent` with value 1 (indicating transition from normal to high alarm state).
- f. Delete the override file to simulate restoration of normal conditions. Verify that a corresponding notification is received with `extremeOptMonTemperatureEvent = 5` (indicating transition from high alarm to normal state).

Expected Results

- Temperature threshold alarm detected and trap generated within one polling cycle
- Trap PDU contains correct `extremeOptMonTemperatureEvent` enumeration
- Recovery trap generated upon condition clearance with appropriate state transition value

3. Method 3: Unqualified Optic Detection

This method verifies that the switch correctly identifies and reports optical transceivers not on the manufacturer's approved compatibility list.

- a. Insert an optical transceiver module not present on the approved qualification list into an available transceiver port.
- b. Monitor the trap receiver for an `extremeIfOpticUnQualified` trap notification. Verify that the trap contains the correct interface index (`ifIndex`) corresponding to the port containing the unqualified transceiver.
- c. Run the **`show media detail`** command at the device management interface. Verify that the Qualified field for the test port displays No or a similar indicator of non-compliance.

Expected Results

- Unqualified transceiver trap received with correct port identification
- Management interface displays unqualified status for the affected port
- System logging records the non-compliance event

4. Method 4: Trap PDU Content Verification

This method provides detailed verification of SNMP trap message format and content. Use an SNMP trap daemon to capture and examine the complete protocol data unit (PDU) for each notification received.

- a. Launch an SNMP trap daemon with verbose output and console logging.

```
snmptrapd -f -Lo -c /etc/snmp/snmptrapd.conf
```

- b. Trigger one or more trap events using the methods described in sections 6.2 or 6.3. Each trap PDU received by the daemon will be displayed in detail to the console.

c. For each received trap, verify the following elements:

- The snmpTrapOID.0 variable binding contains the correct object identifier (OID) corresponding to the trap type.
- All varbinds listed are present in the received PDU. Do not assume optional varbinds if they are listed as mandatory in the specification.
- Varbind values correspond to the condition that triggered the trap. For example, a temperature alarm trap should contain a temperature value exceeding the high alarm threshold.
- SNMP protocol version and community string in the trap message match the configured values (SNMPv2c with the specified trap community string).

Expected Results

- Each trap PDU is successfully received and decoded by the trap daemon
- Trap OID correctly identifies the event type
- All required varbinds are present with accurate values
- Message integrity and proper SNMP encoding confirmed

MIB Support for Optical Transceiver

The implementation adds the EXTREME-OPTICAL-MONITORING-MIB (OID root: .1.3.6.1.4.1.1916.1.65), exposing:

- Three read-only tables: per-module, per-lane, and media-info
- Six trap notification types
- Threshold monitoring in `fwd-hal` triggers these traps

extremeOptMonInfoTable (Per-Port Metrics)

Provides module-level Digital Diagnostic Monitoring (DDM) metrics for each port. All objects are read-only.

Object	OID Suffix	Access	Type	Description
extremeOptMonTemperature	.1.2.1.1	read-only	DisplayString	Temperature (°C) with status suffix
extremeOptMonTxPowerStatus	.1.2.1.2	read-only	INTEGER (1–7)	Tx power alarm status
extremeOptMonTxPower	.1.2.1.3	read-only	DisplayString	Tx power (dBm) with status suffix
extremeOptMonTxPowerVal	.1.2.1.4	read-only	Unsigned32 (μW)	Tx power in microWatts
extremeOptMonRxPowerStatus	.1.2.1.5	read-only	INTEGER (1–7)	Rx power alarm status
extremeOptMonRxPower	.1.2.1.6	read-only	DisplayString	Rx power (dBm) with status suffix

Object	OID Suffix	Access	Type	Description
extremeOptMonRxPowerVal	.1.2.1.7	read-only	Unsigned32 (μW)	Rx power in microWatts
extremeOptMonTxBiasCurrent	.1.2.1.8	read-only	DisplayString	Tx bias (mA) with status suffix
extremeOptMonVoltageValue	.1.2.1.9	read-only	DisplayString	Supply voltage (mV) with status suffix
extremeOptMonTemperatureEvent	.1.2.1.10	accessible-for-notify	INTEGER (1-20)	Temperature threshold transition
extremeOptMonVoltageEvent	.1.2.1.11	accessible-for-notify	INTEGER (1-20)	Voltage threshold transition
extremeOptMonTxPowerEvent	.1.2.1.12	accessible-for-notify	INTEGER (1-20)	Tx power threshold transition
extremeOptMonRxPowerEvent	.1.2.1.13	accessible-for-notify	INTEGER (1-20)	Rx power threshold transition
extremeOptMonTxBiasEvent	.1.2.1.14	accessible-for-notify	INTEGER (1-20)	Tx bias threshold transition

extremeOptMonLaneTable (Per-Lane Metrics)

Provides lane-level metrics for multi-lane optics such as QSFP and QSFP-DD.

Object	OID Suffix	Access	Type	Description
extremeOptMonLaneNum	.1.1.1.1	not-accessible (index)	Unsigned32	Lane number
extremeOptMonLaneTemperature	.1.1.1.2	read-only	SnmpAdminString	Lane temperature (°C)
extremeOptMonLaneTxPowerStatus	.1.1.1.3	read-only	INTEGER (1-7)	Lane Tx power alarm status
extremeOptMonLaneTxPower	.1.1.1.4	read-only	SnmpAdminString	Lane Tx power (dBm)
extremeOptMonLaneTxPowerVal	.1.1.1.5	read-only	Unsigned32 (μW)	Lane Tx power (microWatts)

Object	OID Suffix	Access	Type	Description
extremeOptMonLaneRxPowerStatus	.1.1.1.6	read-only	INTEGER (1-7)	Lane Rx power alarm status
extremeOptMonLaneRxPower	.1.1.1.7	read-only	SnmpAdminString	Lane Rx power (dBm)
extremeOptMonLaneRxPowerVal	.1.1.1.8	read-only	Unsigned32 (μW)	Lane Rx power (microWatts)
extremeOptMonLaneTxBiasCurrent	.1.1.1.9	read-only	SnmpAdminString	Lane Tx bias (mA)
extremeOptMonLaneVoltageValue	.1.1.1.10	read-only	DisplayString	Lane supply voltage (mV)

extremeIfMediaInfoTable (Transceiver Inventory)

Provides transceiver identity information per port.

Object	OID Suffix	Access	Type	Description
extremeIfMediaType	.1.3.1.1	read-only	DisplayString	Form factor type string
extremeIfMediaVendorName	.1.3.1.2	read-only	DisplayString	Vendor full name
extremeIfMediaVersion	.1.3.1.3	read-only	DisplayString	Vendor product version
extremeIfMediaPartNumber	.1.3.1.4	read-only	DisplayString	Vendor part number
extremeIfMediaSerialNumber	.1.3.1.5	read-only	DisplayString	Vendor serial number
extremeIfIsMediaOpticQualified	.1.3.1.6	read-only	INTEGER qualified(1)/unQualified(2)	Optic qualification status

SNMP Trap Notification Types

SNMP Trap Notification Types

The system generates traps when threshold transitions occur or when an unqualified optic is detected.

Trap Name	OID	Trigger Condition	Varbinds
extremeIfTempStateChange	.1.3.6.1.4.1.1916.1.65.0.1	Temperature level changes (ch0 only)	extremeOptMonTemperature, extremeOptMonTemperatureEvent
extremeIfVoltStateChange	.1.3.6.1.4.1.1916.1.65.0.2	Voltage level changes (ch0 only)	extremeOptMonVoltageValue, extremeOptMonVoltageEvent
extremeIfTxPwrStateChange	.1.3.6.1.4.1.1916.1.65.0.3	Tx power level changes (per channel)	extremeOptMonTxPowerStatus, extremeOptMonTxPowerVal, extremeOptMonTxPower, extremeOptMonTxPowerEvent
extremeIfRxPwrStateChange	.1.3.6.1.4.1.1916.1.65.0.4	Rx power level changes (per channel)	extremeOptMonRxPowerStatus, extremeOptMonRxPower, extremeOptMonRxPowerVal, extremeOptMonRxPowerEvent
extremeIfTxBiasStateChange	.1.3.6.1.4.1.1916.1.65.0.5	Tx bias current level changes (per channel)	extremeOptMonTxBiasCurrent, extremeOptMonTxBiasEvent
extremeIfOpticUnQualified	.1.3.6.1.4.1.1916.1.65.0.6	Unqualified optic detected on insertion	ifIndex

*Trap Varbind Detail***Table 6: extremelfTempStateChange**

Varbind OID	Object	Type	Description
1.3.6.1.4.1.1916.1.65.1.2.1.1.<ifIndex>	extremeOptMonTemperature	OctetString	Temperature formatted as "xx.yy Celsius"
1.3.6.1.4.1.1916.1.65.1.2.1.10.<ifIndex>	extremeOptMonTemperatureEvent	Integer (1–20)	Threshold transition code

Table 7: extremelfVoltStateChange

Varbind OID	Object	Type	Description
1.3.6.1.4.1.1916.1.65.1.2.1.9.<ifIndex>	extremeOptMonVoltageValue	OctetString	Voltage formatted as "xxxx.yy mVolts"
1.3.6.1.4.1.1916.1.65.1.2.1.11.<ifIndex>	extremeOptMonVoltageEvent	Integer (1–20)	Threshold transition code

Table 8: extremelfTxPwrStateChange

Varbind OID	Object	Type	Description
1.3.6.1.4.1.1916.1.65.1.2.1.2.<ifIndex>	extremeOptMonTxPowerStatus	Integer (1–7)	Power alarm status enum
1.3.6.1.4.1.1916.1.65.1.2.1.4.<ifIndex>	extremeOptMonTxPowerVal	Gauge32 (μW)	Tx power in microWatts
1.3.6.1.4.1.1916.1.65.1.2.1.3.<ifIndex>	extremeOptMonTxPower	OctetString	Tx power formatted as "x.yy dBm"
1.3.6.1.4.1.1916.1.65.1.2.1.12.<ifIndex>	extremeOptMonTxPowerEvent	Integer (1–20)	Threshold transition code

Table 9: extremelfRxPwrStateChange

Varbind OID	Object	Type	Description
1.3.6.1.4.1.1916.1.65.1.2.1.5.<ifIndex>	extremeOptMonRxPowerStatus	Integer (1–7)	Power alarm

Table 9: extremelfRxPwrStateChange (continued)

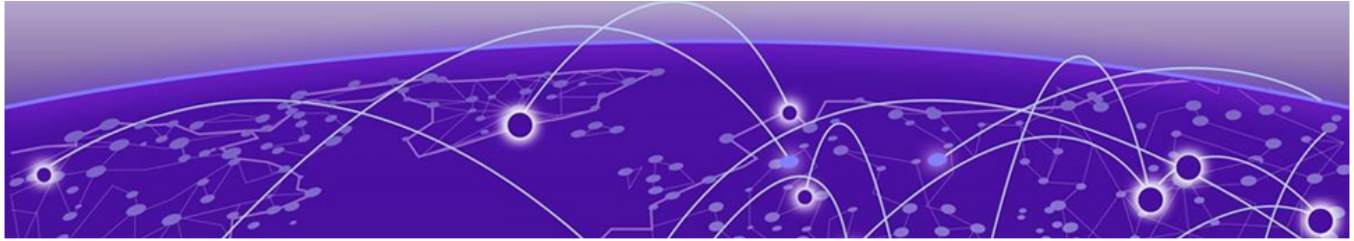
Varbind OID	Object	Type	Description
			status enum
1.3.6.1.4.1.1916.1.65.1.2.1.6.<ifIndex>	extremeOptMonRxPower	OctetString	Rx power formatted as "x.yy dBm"
1.3.6.1.4.1.1916.1.65.1.2.1.7.<ifIndex>	extremeOptMonRxPowerVal	Gauge32 (μW)	Rx power in microWatts
1.3.6.1.4.1.1916.1.65.1.2.1.13.<ifIndex>	extremeOptMonRxPowerEvent	Integer (1–20)	Threshold transition code

Table 10: extremelfTxBiasStateChange

Varbind OID	Object	Type	Description
1.3.6.1.4.1.1916.1.65.1.2.1.8.<ifIndex>	extremeOptMonTxBiasCurrent	OctetString	Tx bias formatted as "x.yy mA"
1.3.6.1.4.1.1916.1.65.1.2.1.14.<ifIndex>	extremeOptMonTxBiasEvent	Integer (1–20)	Threshold transition code

Table 11: extremelfOpticUnQualified

Varbind OID	Object	Type	Description
1.3.6.1.2.1.2.2.1.1.<ifIndex>	ifIndex	Integer	SNMP interface index of the unqualified port



LLDP

[How Network Devices Use LLDP](#) on page 158

[Configuring and Managing LLDP](#) on page 164

[DCBx and IEEE LLDP TLVs](#) on page 169

[LLDP SNMP GET](#) on page 176

How Network Devices Use LLDP

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 you to statically monitor and configure each device in the network.

Using LLDP, network devices 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.

LLDP Operation

LLDP periodically exchanges advertisements with directly connected devices.

The protocol:

1. Transmits LLDP advertisements containing local device information.
2. Receives advertisements from neighboring devices.
3. Stores neighbor information in the LLDP database.
4. Ages out stale entries when advertisements are no longer received.

LLDP supports both physical interfaces and Link Aggregation Groups (LAGs).

LLDP Configuration Scope

LLDP can be controlled globally or per interface.

Global Controls

Configure the following settings globally:

- LLDP enable or disable
- Transmit (TX) enable or disable
- Receive (RX) enable or disable

Interface Controls

Configure the following settings per interface:

- LLDP enable or disable
- TX enable or disable
- RX enable or disable



Note

Global LLDP settings take precedence over interface-level settings. An interface may show LLDP enabled, while LLDP remains operationally disabled due to a global setting.

LLDP Timers

Hello Timer

The Hello Timer controls how frequently LLDP advertisements are transmitted.

LLDP uses this timer to:

- Refresh neighbor information.
- Maintain visibility of directly connected devices.
- Advertise updated device information.

Hold Time

Hold Time specifies how long learned neighbor information remains valid.

If LLDP advertisements are not received before the Hold Time expires:

- The neighbor entry is considered stale.
- The entry is removed from the LLDP database.

SNMP Support

Supported SNMP Versions	Supported Operations	Unsupported Operations
• SNMPv2c • SNMPv3	• GET • GETNEXT • WALK • BULKWALK	• SNMP SET • SNMP notifications and traps

Example Queries

Retrieve the global transmit interval:

```
snmpget -v2c -c public <switch> 1.0.8802.1.1.2.1.1
```

Display per-port administrative status:

```
snmpwalk -v2c -c public <switch> 1.0.8802.1.1.2.1
```

Display all learned neighbors:

```
snmpwalk -v2c -c public <switch> 1.0.8802.1.1.2.1.4.1.1
```

Display TX statistics:

```
snmpwalk -v2c -c public <switch> 1.0.8802.1.1.2.1.2.6.1
```

Display RX statistics:

```
snmpwalk -v2c -c public <switch> 1.0.8802.1.1.2.
```

Log Locations

Use the following locations when troubleshooting LLDP operations:

Service	Location
LLDP	/tmp/tierra/trace/lldp
CLI	/var/log/tierra/cli/
API Gateway	/var/log/tierra/api-gw/
Docker Containers	/var/log/containers/
Kubernetes Pods	/var/log/pods/

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 device 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 OS ONE, 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 devices 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 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.

Prerequisites

Before using LLDP, ensure that:

- LLDP is globally enabled on the device.
- Interfaces participating in LLDP are operational.
- Management services are configured when management address advertisement is required.
- Neighboring devices support LLDP.
- For Fabric Extend deployments, both devices share the same authentication password used for FE TLV validation.

Supported Platforms

LLDP is supported on the following platforms:

- 8720-32C
- 8520-48Y
- 8520-48XT
- 9250-32C
- 9150-48Y
- 9150-48XT
- 9740-40C
- 9740-80C
- IPE-40AXV2
- IPE-420AX
- IPE-2200F
- vOS (Virtual OS ONE)

Limitations

The following limitations apply:

- LLDP operates only on point-to-point links.
- User-defined configuration of optional TLVs is not supported.
- LLDP SNMP support is read-only.
- SNMP notifications are not supported.
- Several LLDP-MIB extensions and LLDPv2 MIB objects are outside the supported scope.

Understanding the Default LLDP

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

Table 12: 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

Disable 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 Information
```

Chassis ID System Name	Local-Port	Dead Intvl	Rem Life	Remote Port-ID	Remote Port Descr
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 detailed information about Ethernet interface 0/1-2 on neighbor devices discovered using LLDP:

```
show lldp neighbors interface ethernet 0/1-2
```

Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

LLDP Interface: ethernet 0/1
 Chassis id: 88f0.31f9.a340
 Port id: ethernet 1/10
 Local Port id: ethernet 0/1
 Port Description: ethernet 1/10
 System Name: Lab switch1
 System Description: "ExtremeXOS (Summit) version
 31.1.1.4 v311b4 by release-manager on Fri Feb 13 16:30:06 EST 2021"
 Time remaining: 91 seconds
 System Capabilities: B, R
 Enabled Capabilities: B, R
 Management Address: 10.77.143.43
 Management Address IPV6: not advertised

LLDP Interface: ethernet 0/2
 Chassis id: 88f0.31f9.a341
 Port id: ethernet 1/11
 Local Port id: ethernet 0/2
 Port Description: ethernet 1/11
 System Name: Lab switch 2
 System Description: Cisco Nexus Operating System (NX-OS) Software 9.2(1) TAC support:
<http://www.cisco.com/tac> Copyright (c) 2002-2018, Cisco Systems, Inc. All rights reserved.


```
Time remaining: 85 seconds
System Capabilities: B, R
Enabled Capabilities: B, R
Management Address: 10.77.143.43
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.

Clear 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 0/2
```

This example clears the LLDP statistics for Ethernet interface 0/2.

2. Clear the LLDP statistics for all Ethernet interfaces.

```
device# clear counters lldp interface ethernet all
```

3. Clear all the LLDP statistics.

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

DCBx and IEEE LLDP TLVs

DCBx extends LLDP by using organizationally specific TLVs to exchange and negotiate lossless Ethernet and QoS parameters between directly connected peers. This

mechanism helps ensure that both ends of a link apply compatible scheduling, bandwidth allocation, and flow-control behavior in converged Ethernet environments.

The IEEE LLDP TLVs described in this guide provide peer visibility into VLAN, MTU, and link aggregation information. These TLVs are informational and improve operational awareness and interoperability.

This topic is intended for network administrators and operators who deploy or manage DCBx and LLDP-based interface advertisement in Extreme OS ONE environments.

Use this topic if your network requires:

- QoS coordination between directly connected Ethernet peers.
- Lossless Ethernet behavior for selected traffic classes.
- Application-to-priority mapping for traffic classification.
- LLDP-based visibility into VLAN, MTU, and link aggregation parameters.

DCBx

DCBx Overview

DCBx is a discovery and negotiation mechanism that runs over LLDP and exchanges organizationally specific Type 127 TLVs between directly connected peers. It is used in converged Ethernet networks to advertise local capabilities and to derive compatible operational settings between neighbors.

DCBx negotiates the following capabilities (QoS features):

- **ETS (Enhanced Transmission Selection)** for bandwidth allocation across traffic classes.
- **PFC (Priority Flow Control)** for lossless delivery on selected priorities.
- **APP (Application Priority)** for mapping applications to IEEE 802.1p priorities.
- **ETS Recommendation** for advertising a preferred ETS model to a peer.

In converged Ethernet networks, different traffic types can have different latency and loss requirements. DCBx reduces the need for manual symmetric configuration by automating peer capability discovery and exchanging QoS parameters through LLDP.

DCBx provides the following advantages:

- Automatic peer capability discovery.
- Exchange of ETS, PFC, and APP parameters over LLDP.
- Negotiation logic based on feature willingness.
- Detection of incompatible QoS settings.
- Publication of negotiated operational state to QoS subsystems

DCBx Protocol Support

This implementation supports the following DCBx variants:

- **IEEE 802.1Qaz** with support for ETS Configuration, ETS Recommendation, PFC Configuration, and APP Priority TLVs.
- **CEE (Intel DCBx)** in receive-only mode, with decode and publish support for Control, ETS/Priority Groups, PFC, and APP TLVs.

Version detection is automatic based on the received OUI. IEEE 802.1Qaz uses OUI 00-80-C2, and legacy CEE DCBx uses OUI 00-1B-21.



Note

CEE DCBx support is limited to receive-side decode and publish. This implementation does not advertise CEE DCBx TLVs.

DCBx Feature Summary

ETS

ETS allocates bandwidth across traffic classes while allowing strict-priority forwarding where required. It enables fair bandwidth sharing during congestion and supports up to eight traffic classes in this implementation.

The ETS configuration model includes:

- Maximum traffic classes.
- Priority-to-traffic-class assignment.
- Bandwidth allocation per traffic class.
- Transmission Selection Algorithm assignment per traffic class.
- A willing bit to indicate whether peer recommendations can influence operational state.

RFC

PFC provides lossless Ethernet behavior on a per-priority basis. It uses an enable bitmap to identify which IEEE 802.1p priorities are lossless and includes a capability limit that defines how many priorities can be enabled simultaneously.

The PFC state includes:

- Willing bit.
- MACsec bypass capability field, which is defined but not supported in this implementation.
- PFC capability (PFCCap).
- PFC enable bitmap.

APP

APP maps higher-layer application traffic to Ethernet priorities so that traffic is classified correctly before ETS and PFC take effect. Each APP entry identifies a priority, a selector type, and an application identifier.

Supported APP selector types include:

- EtherType.
- TCP port.
- UDP port.
- TCP or UDP port.
- DSCP.

A maximum of 32 APP entries is supported.

ETS Recommendation

ETS Recommendation uses the same payload format as the ETS Configuration TLV, but it is advisory. It allows a device to recommend a preferred priority-to-traffic-class model and bandwidth allocation to a peer. In this implementation, only receive-side decode and publish are supported for ETS Recommendation.

DCBx Limitations

Be aware of the following implementation limits:

- DCBx depends on LLDP being enabled.
- DCBx is configured per interface, not globally.
- A maximum of **8 traffic classes** is supported for ETS.
- **CBS (Credit-Based Shaper)** is not supported.
- **MACsec bypass capability** is not supported for PFC.
- APP compatibility requires overlapping priority mapping for common applications.
- ETS Recommendation support is limited to **receive-side decode and publish**.
- CEE DCBx support is limited to **receive-side decode and publish**.

LLDP Dependency and Operational Model

All DCBx information is carried within LLDP Type 127 organizationally specific TLVs. LLDP must therefore be enabled for both transmission and reception.

DCBx is configured on a **per-interface basis** and is not applied globally.

Transmission behavior

When DCBx is enabled on an interface, the device transmits DCBx TLVs only if:

- DCBx is enabled.
- At least one DCBx feature configuration is present, such as ETS, PFC, APP, or ECN.

If DCBx is enabled but no relevant DCBx feature configuration exists, no DCBx TLVs are advertised on the interface.

Reception behavior

The device always listens for incoming DCBx TLVs through LLDP, regardless of whether DCBx is enabled locally. Received TLVs are continuously decoded and stored in RX state. When DCBx is enabled, the preserved RX state is used for initial negotiation.

RoCEv2 Example Use Cases

The specification includes RoCEv2-oriented examples for PFC and APP mapping.

PFC example

A typical RoCEv2 PFC configuration advertises:

- Willing bit cleared so peer values are not adopted automatically.
- PFC capability of **8**.
- PFC enable bitmap of 0x08, which enables PFC for priority **3** only.

APP example

A typical RoCEv2 APP model includes entries such as:

- UDP port **4791** mapped to priority **3** for RoCEv2 data.
- DSCP **48** mapped to priority **6** for congestion notification traffic.
- DSCP **26** mapped to priority **3** for RoCE data.

Operational State and Notifications

The implementation maintains and publishes state for both transmitted and received DCBx information. The state model includes:

- Local DCBx configuration state.
- Peer-received TLV state.
- Negotiated operational state.
- Feature-level status such as operational mode, synchronization status, and error state.

Notifications are published per interface and include:

- Peer MAC address and version.
- Negotiated operational configuration.
- Feature-level state information.

A peer-lost notification is generated when the LLDP TTL expires.

Configure DCBx

Before enabling DCBx, ensure the following conditions are met:

- LLDP is enabled on the target interface.
- The required DCBx feature configuration is defined for the interface.
- You understand that DCBx negotiation occurs per interface and depends on peer advertisements and local willingness settings.

Follow this procedure to configure DCBx.

Enable DCBx.

DCBx is enabled from the LLDP interface configuration hierarchy by using the documented CLI path:

```
ldp → dcbx → enable
```

Enabling this control activates DCBx transmit behavior when feature configuration is present and enables negotiation based on the locally stored and received TLV data.

Expected behavior after DCBx is enabled:

- The interface can advertise ETS, PFC, APP, and ECN-related organizationally specific TLVs through LLDP.
- Received peer TLVs can be used in negotiation processing.
- Peer and operational state can be published through the management and state model.

DCBx Validation Rules

Use the following rules when preparing or validating configuration:

ETS Validation

- **MaxTCs** must be in the range **1 through 8**.
- The sum of ETS bandwidth values must equal **100**.
- TSA values must be valid and must use supported scheduling modes.

PFC validation

- The number of priorities enabled for PFC must not exceed **PFCCap**.

APP validation

- No more than **32 entries** are supported.
- The selector must be valid.
- Priority values must be in the range **0**

IEEE LLDP TLVs

IEEE LLDP TLV overview

The IEEE LLDP TLVs supported in this implementation advertise local VLAN and interface information to directly connected peers. These TLVs help identify configuration mismatches and improve interoperability, but they do not configure the receiving device.

Supported IEEE TLVs include:

- **VLAN Name TLV.**
- **Port VLAN ID (PVID) TLV.**
- **Maximum Frame Size TLV.**
- **Link Aggregation TLV.**

These TLVs provide peer visibility into:

- VLAN identifiers and VLAN names.
- The default VLAN used for untagged ingress traffic.

- The maximum supported frame size on the interface.
- Whether the interface supports and participates in link aggregation.

They improve operational awareness and help identify mismatches that could otherwise result in traffic drops or interoperability issues.

IEEE TLV Descriptions

VLAN Name TLV

The VLAN Name TLV advertises VLAN identifiers together with human-readable VLAN names. It is intended for visibility and management consistency across peers. Multiple VLAN entries can be advertised. This TLV is informational and does not participate in DCBx negotiation. [\[DCBX++Azucification | Word\]](#)

Port VLAN ID (PVID) TLV

The PVID TLV advertises the default VLAN applied to untagged ingress traffic on the port. In this implementation, default VLAN functionality is not supported on the documented TD3 devices, so the system advertises a PVID value of 0 to indicate that the default VLAN is unknown or unsupported. [\[DCBX++Azucification | Word\]](#)

Maximum Frame Size TLV

The Maximum Frame Size TLV advertises the maximum Ethernet frame size supported by the interface. It is used to improve interoperability and MTU visibility between peers.

Link Aggregation TLV

The Link Aggregation TLV advertises whether the interface supports link aggregation, whether aggregation is currently enabled, and the aggregation group port identifier when applicable. It provides peer visibility into LAG membership and status.

IEEE TLV Limitations

- IEEE TLVs are **informational only**. They do not configure peer devices.
- VLAN TLVs do not create or modify VLAN configuration on the peer.
- The Maximum Frame Size TLV does not enforce MTU values on the peer.
- The Link Aggregation TLV does not control LAG state.
- The PVID TLV is limited because default VLAN functionality is not supported in the documented implementation.
- Operation is per interface only and depends on LLDP transmit and receive capability.

Configure IEEE TLVs

Before enabling IEEE TLV advertisement, ensure that:

- LLDP is enabled on the interface.
- The interface has the relevant local state available for VLAN, MTU, or aggregation reporting.

Follow this procedure to configure IEEE TLVs.

Enable IEEE TLVs.

IEEE TLV advertisement is enabled from the LLDP interface hierarchy by using the documented CLI path:

```
lldp → ieee_tlvs → enable
```

When enabled, the interface advertises the supported IEEE TLVs through LLDP.

- **Reception behavior:** IEEE TLV reception remains active through LLDP regardless of whether `ieee_tlvs` is enabled. Received TLVs are decoded and stored in the receive-state database and are updated on each LLDPDU reception.
- **Transmission behavior:** With IEEE TLVs enabled, the system advertises:
 - VLAN information.
 - VLAN Name.
 - Port VLAN ID (PVID).
 - Link Aggregation information.
 - Maximum Frame Size.

LLDP SNMP GET

The LLDP SNMP GET feature provides read-only SNMP access to LLDP information maintained by the OneOS LLDP service. This implementation exposes a subset of the standard IEEE/IETF LLDP-MIB and allows network management systems to monitor and retrieve LLDP configuration, interface status, statistics, local port information, and neighbor discovery using standard SNMP retrieval operations while maintaining a read-only management model.

The feature supports monitoring and inventory use cases and does not permit modification of LLDP parameters through SNMP.

Supported SNMP Versions and Operations

Supported SNMP Versions	Supported Operations	Unsupported Operations
• SNMPv2c • SNMPv3	• <code>snmpget</code> • <code>snmpgetnext</code> • <code>snmpwalk</code> • <code>snmpbulkwalk</code> • LLDP Remote Neighbor Information • LLDP Remote Management Address Information	• SNMP SET requests • SNMP notifications and traps • LLDP Statistics global scalars • <code>lldpRemUnknownTLVTable</code>, <code>lldpRemOrgDefInfoTable</code> • <code>lldpV2-MIB</code> and <code>lldpXdot{1,3}</code>* extensions Unsupported objects return <code>noSuchInstance</code>.

LLDP MIB Support

The feature exposes data from the standard LLDP-MIB. Unsupported MIB objects return `noSuchInstance`.

Item	Value
MIB Module	LLDP-MIB
MIB OID	1.0.8802.1.1.2
Root OID	1.0.8802.1.1.2.1
Access Mode	Read-only

Unsupported LLDP Objects

The following LLDP-related objects are not supported:

- `IldpRemUnknownTLVTable`
- `IldpRemOrgDeflInfoTable`
- `IldpLocManAddrTable`
- `IldpConfigManAddrTable`
- `IldpRemManAddrTable`
- LLDPV2-MIB
- `IldpXdot1*`
- `IldpXdot3*`

Requests to unsupported objects return `noSuchInstance`.

Limitations

Be aware of the following implementation constraints:

- SNMP access is read-only.
- SNMP SET operations are not supported.
- LLDP notifications are not supported.
- Only a subset of LLDP-MIB objects is exposed.
- Unsupported objects return `noSuchInstance`.
- A single management address is maintained per discovered neighbor.

Retrieving Global LLDP Information

The following examples use SNMP v2c with the community string `public`. Replace `<switch>` with the hostname or IP address of your device, and substitute your own community string or v3 credentials.

View Global LLDP Configuration

Use SNMP GET operations to retrieve global LLDP configuration parameters.

Retrieve LLDP Transmit Interval

Example

```
snmpget -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.1.1.0
```

Returns

The configured LLDP transmit interval in seconds.

Valid range:

- 5 to 32768 seconds

Object

Name	OID
lldpMessageTxInterval	1.0.8802.1.1.2.1.1.1.0

Retrieve LLDP Hold Multiplier

Example

```
snmpget -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.1.2.0
```

Returns

The LLDP message hold multiplier.

Valid range:

- 2 to 10

Object

Name	OID
lldpMessageTxHoldMultiplier	1.0.8802.1.1.2.1.1.2.0

View LLDP Port Configuration

The `lldpPortConfigTable` provides LLDP configuration details for individual interfaces.

Retrieve Port Configuration Information

Example

```
snmpwalk -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.1.6.1
```

Table Index: ifIndex

Available Information

- Port number
- Administrative status
- Enabled transmit and receive functions
- Transmitted TLV settings

View LLDP Statistics

The LLDP statistics tables provide transmit and receive counters for each interface.

View Transmit Statistics

Example

```
snmpwalk -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.2.6.1
```

Available Counters: Total transmitted LLDP frames

View Receive Statistics**Example**

```
snmpwalk -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.2.7.1
```

Available Counters

- Discarded frames
- Frame errors
- Received frames
- Discarded TLVs
- Unrecognized TLVs
- Age-out count

**Note**

The Age-out counter returns a constant value of 0.

View Local LLDP Port Information

The `lldpLocPortTable` provides information about local switch interfaces participating in LLDP.

Retrieve Local Port Information**Example**

```
snmpwalk -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.3.7.1
```

Available Information

- Port ID subtype
- Port ID
- Port description

Port ID Subtype

The implementation returns:

```
5 (interfaceName)
```

View Learned LLDP Neighbors

The `lldpRemTable` contains information learned from neighboring LLDP devices.

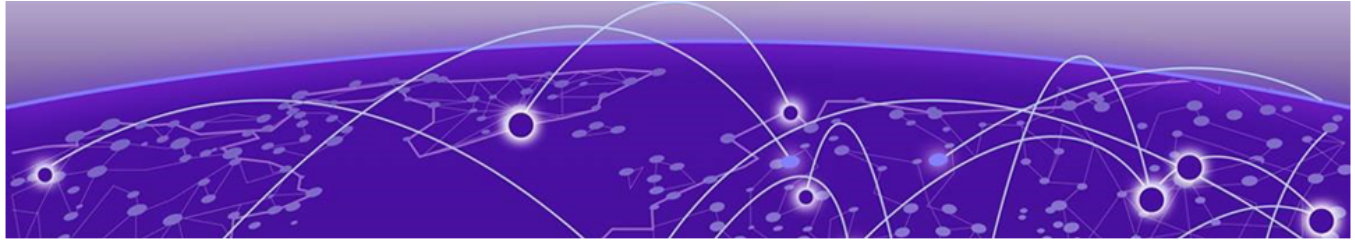
Retrieve Neighbor Information**Example**

```
snmpwalk -v2c -c public <switch-ip> 1.0.8802.1.1.2.1.4.1.1
```

Available Information

- Chassis ID
- Chassis ID subtype
- Port ID

- Port ID subtype
- Port description
- System name
- System description
- Supported capabilities
- Enabled capabilities
- Neighbor time mark



gNMI and gNOI

[How gNMI and gNOI Protocols Manage Networks](#) on page 181
[gRPC Network Management Interface \(gNMI\) Service](#) on page 184
[gRPC Network Operational Interface \(gNOI\) Service](#) on page 194
[Configure gRPC Server](#) on page 195
[Data Model and Northbound Interface](#) on page 196
[Storage of List Key Values](#) on page 197
[Special Characters in Leaf Names and List Key Values](#) on page 197
[Inband Management Support](#) on page 201
[CLI Commands](#) on page 201
[Event Log Messages](#) on page 202
[gNMI Authentication and Encryption](#) on page 202
[Import gNMI Client CA Root Certificate](#) on page 203
[Security](#) on page 203

How gNMI and gNOI Protocols Manage Networks

Extreme OS ONE 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 Protocol

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.

Limitations for Subscription Paths

- Minimum Sampling Intervals for Stream Subscriptions

The following paths have minimum sampling interval requirements and you cannot set the intervals below the specified values:

```
LowestIntervalPerPath: <Time in seconds>
"/interfaces": 40
"/interfaces/interface[name=*]": 40
"/interfaces/interface": 40
"/bfd/interfaces/interface[id=*]": 30
"/bfd/interfaces/": 30
"/network-instances": 50
"/network-instances/network-instance": 50
"/network-instances/network-instance[name=*]": 50
```

- Unsupported On-Change Stream Subscriptions

The following paths do not support gNMI On-Change subscriptions. To retrieve data for these paths and their subordinate nodes, use gNMI Once or Sample subscriptions instead of querying the parent path. Explicit queries are required.

```
- /network-instances/network-instance/protocols/protocol/bgp/rib
- /network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/
timers/state
- /network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/afi-
safis/afi-safi/state/prefixes
- /network-instances/network-instance/protocols/protocol/bgp/global/afi-safis/afi-
safi/state/total-prefixes
- /network-instances/network-instance/protocols/protocol/bgp/global/afi-safis/afi-
safi/state/total-paths
- /network-instances/network-instance/tables-network/ipv4-unicast
- /network-instances/network-instance/tables-network/ipv6-unicast
- /network-instances/network-instance/tables-network/next-hop-groups
- /network-instances/network-instance/tables-network/next-hops
- /network-instances/network-instance/tables-network/ethernet
- /network-instances/network-instance/tables-network/ipv4-neighbor
- /network-instances/network-instance/tables-network/ipv6-neighbor
- /system/global/l3/tables-network-summary/address-family-stats
```

- Unsupported Get RPC Path Requests

The following paths are not supported on gNMI Get:

```
GetUnsupportedPaths:
- /network-instances/network-instance/protocols/protocol/bgp/rib
- /network-instances/network-instance/tables-network
- /system/global/l3/tables-network-summary
- /network-instances
- /interfaces
- /acl
- /qos
- /qos/maps/traffic-class-maps/
```

gNOI Protocol

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 OS ONE supports gNMI version 0.10.0.

Use the Capabilities RPC to discover available encodings on the target device. The response advertises both PROTO (default protobuf encoding) and JSON_IETF as supported options. Query capabilities first, then request the encoding that matches

the target's advertised support. 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 OS ONE 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 **show grpc-server gnmi capabilities** command. This command provide details of supported YANG data models, gNMI version, and encoding formats.

**Note**

The **show grpc-server gnmi statistics** command provide details of active stream subscriptions, including client information, subscription mode, start time, and path details.

Get RPC

Retrieve data in JSON IETF format by setting the encoding field in the GetRequest message to JSON_IETF. If the requested encoding is not specified or unsupported, the system automatically defaults to PROTO encoding without returning an error. This lenient fallback behavior ensures backward compatibility.

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

Subscribe to streaming telemetry in JSON IETF format by setting the encoding field in the SubscriptionList message to **JSON_IETF**. All notifications in the subscription use the requested encoding. Unlike Get RPC, Subscribe RPC validates the requested encoding against device capabilities and returns an error if unsupported encoding is requested.

```
Example: SubscribeRequest{subscribe: SubscriptionList{..., encoding:
JSON_IETF}}
```

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.

gNMI JSON-IETF Encoding Support

The gNMI service supports **JSON_IETF** encoding for gNMI **Get** and **Subscribe** operations, in addition to the default **PROTO** encoding. JSON_IETF follows **RFC 7951** and allows clients to receive YANG-modeled data in a standardized, human-readable JSON format.

JSON_IETF encoding provides the following advantages:

- **Human readability:** Data is returned in standard JSON with YANG leaf names as keys, which simplifies debugging.
- **Standards compliance:** Complies with **IETF RFC 7951**, ensuring vendor-neutral interoperability across platforms
- **Universal tooling support:** Compatible with all standard JSON parsers, validators, and formatters in any language
- **Schema-independent consumption:** Inspect and parse the JSON output without requiring YANG schema knowledge upfront for basic analysis and troubleshooting.
- **Easy Manual Inspection:** Craft and examine payloads using any text editor

- **Improved troubleshooting:** JSON is easier to inspect manually in logs, files, and editors without decoding — streamlines troubleshooting and compliance auditing
- **Efficient text-based file output:** Produces approximately 55% smaller files than prototext format for text-based telemetry storage

**Note**

While binary protobuf remains efficient for wire transport and binary-oriented workflows, JSON_IETF is better suited for human-readable file output and log inspection.

Supported Platforms

JSON_IETF encoding support is available on **all platforms running the Extreme OS ONE Switching and Routing application**.

Supported Operations

The following gNMI RPCs are enhanced to support JSON_IETF encoding:

- Capabilities RPC
- Get RPC
- Subscribe RPC

Large Payload Handling

For **Subscribe** RPCs, payloads larger than **3.5 MB** are automatically split into multiple notification messages to avoid exceeding the **4 MB gRPC message size limit**. This splitting behavior is consistent for both **PROTO** and **JSON_IETF** encodings.

All split notifications that belong to the same response carry the **same timestamp**. Clients can use the shared timestamp to identify and reassemble notifications that belong to the same dataset.

Important behavior

- Automatic payload splitting applies to **Subscribe RPC only**.
- The **Get RPC** is intended for relatively small datasets and may fail with a gRPC message size error if the response exceeds the limit.
- For large-scale data retrieval, use Subscribe RPC instead of Get.

Best Practices

Use the following recommendations when working with JSON_IETF encoding:

- Run **Capabilities** before requesting a specific encoding.
- Use **JSON_IETF** when output readability, text-based logging, or generic JSON tooling is required.
- Use **PROTO** when binary-oriented workflows or default client behavior are preferred.
- Use **Subscribe** instead of **Get** when the returned dataset might exceed the gRPC message size limit.
- For split notification handling, correlate data by **timestamp** and **prefix**, then reconstruct the hierarchy by using the **update path**.

Use JSON_IETF with Get RPC

The **Get** RPC supports JSON_IETF encoding and allows clients to retrieve data in compliance with **RFC 7951**.

To request JSON_IETF encoding, set the **encoding** field in the **GetRequest** message to JSON_IETF.

Get RPC behavior

The Get RPC uses **lenient fallback** behavior:

- If the client explicitly requests JSON_IETF, the service returns data in JSON_IETF format.
- If no encoding is specified, the service defaults to **PROTO**.
- If an unsupported encoding is requested, the service defaults to **PROTO**.
- The Get RPC does **not** validate the requested encoding against Capabilities before applying fallback behavior.

Example: Get with PROTO

The following example requests data from the specified path by using PROTO encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> get --path <path> --  
encoding proto
```

Example: Get with JSON_IETF

The following preceding example requests the same data in JSON_IETF format.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> get --path <path> --  
encoding json_ietf
```

Use JSON_IETF with Subscribe RPC

The **Subscribe** RPC supports JSON_IETF encoding for streaming telemetry and state updates. To request JSON_IETF encoding, set the **encoding** field in the **SubscriptionList** to JSON_IETF.

Subscribe RPC Behavior

The Subscribe RPC uses **strict validation** behavior:

- If the client requests JSON_IETF and the encoding is supported, the subscription proceeds normally.
- If the client requests an unsupported encoding, the subscription fails with an explicit error.
- The requested encoding is validated against the target's advertised capabilities.

Example: SAMPLE subscription with PROTO

The following example starts a SAMPLE stream subscription using PROTO encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode stream \  
  --stream-mode sample \  
  --sample-interval <sampleinterval> \  
  --encoding proto
```

Example: SAMPLE subscription with JSON_IETF

The following example starts a SAMPLE stream subscription using JSON_IETF encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode stream \  
  --stream-mode sample \  
  --sample-interval <sampleinterval> \  
  --encoding json_ietf
```

Example: ON_CHANGE subscription with PROTO

The following example starts an ON_CHANGE streaming subscription using PROTO encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode stream \  
  --stream-mode on-change \  
  --encoding proto
```

Example: ON_CHANGE subscription with JSON_IETF

The following example starts an ON_CHANGE streaming subscription using JSON_IETF encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode stream \  
  --stream-mode on-change \  
  --encoding json_ietf
```

Example: ONCE subscription with PROTO

The following example retrieves a one-time dataset by using PROTO encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode once \  
  --encoding proto
```

Example: ONCE subscription with JSON_IETF

The following example retrieves a one-time dataset by using JSON_IETF encoding:

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe \  
  --path <path> \  
  --mode once \  
  --encoding json_ietf
```

Verify Supported Encoding

Follow this procedure to verify that the target supports JSON_IETF before sending operational requests.

1. Run the Capabilities command.

This command retrieves the encodings and capabilities advertised by the target gNMI service.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> capabilities
```

2. Confirm the encoding list.

Verify that the Capabilities response includes **JSON_IETF** in addition to **PROTO**.

3. Use the desired encoding.

After confirming support, send the Get or Subscribe request with the `--encoding json_ietf` option.

4. You can request a specific encoding using the `--encoding` flag:

- a. Check capabilities.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> capabilities
```

- b. Get with PROTO encoding.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> get --path <path> --encoding proto
```

- c. Get with JSON IETF encoding.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> get --path <path> --encoding json_ietf
```

- d. SAMPLE subscription with PROTO,

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
  --path <path>
  --mode stream
  --stream-mode sample
  --sample-interval <sampleinterval>
  --encoding proto
```

- e. SAMPLE subscription with JSON IETF.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
  --path <path>
  --mode stream
  --stream-mode sample
  --sample-interval <sampleinterval>
  --encoding json_ietf
```

- f. ON_CHANGE subscription with PROTO.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
  --path <path>
  --mode stream
  --stream-mode on-change
  --encoding proto
```

- g. ON_CHANGE subscription with JSON IETF.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
  --path <path>
  --mode stream
  --stream-mode on-change
  --encoding json_ietf
```

h. ONCE subscription with PROTO.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
--path <path>
--mode once
--encoding proto
```

i. ONCE subscription with JSON IETF.

```
gnmic -a <address>:443 --tls-ca <tls-cert.pem> subscribe
--path <path>
--mode once
--encoding json_ietf
```

Notification Splitting

When a large subscription response is split across multiple notifications, the structure follows these rules:

- The **prefix** remains the same across all notifications in the split response.
- In the first notification, the **update path** can be empty to represent the top-level data under the prefix.
- As the system splits data deeper into the JSON hierarchy, the **update path** becomes populated to identify the nested location of the returned data.
- Every notification in the split response uses the **same timestamp**.

Client Recommendations

When you process split notifications, use the following approach:

- Group notifications by **timestamp** and **prefix**. Notifications with the same timestamp and prefix belong to the same data fetch.
- Combine **prefix** and **update path** to determine where each JSON fragment belongs in the hierarchy.

JSON_IETF Notification Examples

Example 1: Notification without Splitting

The following example shows a single notification that contains a complete JSON_IETF payload for an interface:

```
update: {
  timestamp: 1771584406898218445
  prefix: {
    elem: {
      name: "interfaces"
    }
    elem: {
      name: "interface"
      key: {
        key: "name"
        value: "ethernet 0/1"
      }
    }
  }
}
update: {
  path: {}
  val: {
```



```

    json_ietf_val: "{\"config\":{\"enabled\":false},\"ethernet\":{\"state\":{\"fec\":{\"fec-mode\":\"disabled\"},\"hw-mac-address\":\"00:16:3e:6f:fc:08\"},\"interfaces-auto-negotiation\":false,\"last-local-fault\":\"0\",\"last-remote-fault\":\"0\",\"link-fault-signaling\":true,\"link-fault-status\":\"No Fault\",\"local-fault-count\":\"0\",\"port-speed\":\"SPEED_100GB\",\"remote-fault-count\":\"0\"}},\"name\":\"ethernet 0/1\",\"state\":{\"admin-status\":\"DOWN\",\"counters\":{\"if-mib-counter\":{\"if-counter-discontinuity-time\":\"0\"}},\"description\":\"Ethernet 0/1\",\"enabled\":false,\"if-mib\":{\"if-last-change\":\"1m11s\"},\"ifindex\":16777248,\"max-supported-speed\":[\"SPEED_100GB\"],\"mtu\":9216,\"name\":\"ethernet 0/1\",\"oper-down-reason\":\"\",\"oper-status\":\"DOWN\",\"type\":\"gigabitEthernet\"}}}"
  }
}

```

Example 2: Notification Sequence with Splitting

The following examples show how a large response can be split into multiple notifications that share the same timestamp and prefix:

- Notification 1

This notification returns top-level data below the prefix, indicated by an empty update path.

```

update: {
  timestamp: 1771584977934564191
  prefix: {
    elem: {
      name: "interfaces"
    }
    elem: {
      name: "interface"
      key: {
        key: "name"
        value: "ethernet 0/1"
      }
    }
  }
  update: {
    path: {}
    val: {
      json_ietf_val: "{\"config\":{\"enabled\":false},\"name\":\"ethernet 0/1\"}"
    }
  }
}

```

- Notification 2

This notification returns the portion of the payload that belongs to the state subtree.

```

update: {
  timestamp: 1771584977934564191
  prefix: {
    elem: {
      name: "interfaces"
    }
    elem: {
      name: "interface"
      key: {
        key: "name"
        value: "ethernet 0/1"
      }
    }
  }
}

```

```

update: {
  path: {
    elem: {
      name: "state"
    }
  }
  val: {

json_ietf_val: "{\"if-mib\":{\"if-last-change\":\"1m11s\"},\"ifindex\":16777248,\"max-
supported-speed\":[\"SPEED_100GB\"],\"mtu\":9216,\"name\":\"ethernet 0/1\",\"oper-down-
reason\":\"\"}"
  }
}

```

- Notification 3

This notification returns a deeper portion of the hierarchy under ethernet/state.

```

update: {
  timestamp: 1771584977934564191
  prefix: {
    elem: {
      name: "interfaces"
    }
  }
  elem: {
    name: "interface"
    key: {
      key: "name"
      value: "ethernet 0/1"
    }
  }
}
update: {
  path: {
    elem: {
      name: "ethernet"
    }
    elem: {
      name: "state"
    }
  }
  val: {
    json_ietf_val: "{\"port-speed\":\"SPEED_100GB\"}"
  }
}

```

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:

```
device# certificate-manager generate ssl-profile-id ssl-reserved-generated certificate-extension
san 1.1.1.1 (1.1.1.1 is device ip)
device# certificate-manager import ssl-profile-id ssl-reserved-generated ca-certificate
protocol scp host 1.1.1.1 certificate /tmp/cert.crt user user1 password pass1 vrf mgmt-vrf
device#
```

Or import a custom certificate and associate it with the user defined ssl-profile ID.

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. You can use clients like gnmic for gNMI set, subscribe, and on-change operations.

3. To enable mutual TLS, import the application certificate and associate it with the gRPC server instance using the certificate manager.

```
device# certificate-manager import ssl-profile-id test_cert_id app-certificate
protocol scp host 1.1.1.1 certificate /home/user/gRPCcert.pem key /home/user/gRPCcert.key user username password password
```

4. Import the CA certificate used to generate the app certificate, using the same SSL profile ID as in Step 3.

```
device# certificate-manager import ssl-profile-id test_cert_id ca-certificate protocol
scp host 1.1.1.1 certificate /home/user/root.pem user username password password
```

5. 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)#
```

6. 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
```

7. 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 OS ONE 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.

Special Characters in Leaf Names and List Key Values

Beginning with OS ONE 22.2.3.0, the previously enforced ASCII character restrictions for **leaf names** and **list key values** have been removed. You can now configure and operate with a broader range of ASCII characters in gNMI paths and YANG data model values.

In general, **no ASCII character restrictions apply to leaf names or list key values** unless explicitly imposed by a specific YANG module or feature implementation.

This enhancement applies to user-defined identifiers such as:

- VRF names
- ACL names
- QoS profile names
- QoS map and remark-map names
- Traffic-map names
- Class-map names
- Route-map names
- Community-list names
- Policy names
- Tunnel interface names
- Other configurable objects that use string-based leaf values or list keys



Note

- Special characters are preserved as entered and are returned correctly by supported APIs.
- Escaping is required only when constructing gNMI path strings and does not alter the underlying configured value.
- Apart from documented module-specific constraints and the reserved wildcard character (*), no ASCII character restrictions are enforced for leaf names or list key values.

Printable ASCII Character Support

Generic string-type configuration fields support printable ASCII characters as the standard input set, unless a specific YANG module or feature defines additional restrictions.

The following examples are valid values for generic string fields:

- Ticket #4940233

- Priority >4
- ops]east!@#\$\$%^&

The value *****SPECIAL [FW]***** is not accepted when used as a list key value because the asterisk (*) remains a reserved wildcard character for gNMI path matching. Module-specific restrictions continue to take precedence.



Note

- Description fields retain the same behavior as in OS ONE 22.2.2.0.
- Printable ASCII characters are preserved as entered and returned unchanged through CLI, gNMI, and internal databases.
- Supported values remain subject to any validation rules defined by the applicable YANG module.

CLI Example

- Using Printable ASCII Characters in a Mirror Session Name

```
48xt# conf t
48xt(config)# mirror ops]east!@#$$%^&
48xt(config-mirror)# destination interface ethernet 0/1
48xt(config-mirror)# description "mirror session named ops]east!@#$$%^&"
48xt(config-mirror)# end
```

- Verify the running configuration.

```
48xt# show running-config mirror
mirror ops]east!@#$$%^&
    description mirror session named ops]east!@#$$%^&
    destination interface ethernet 0/1
!
```

- Verify the operational state.

```
48xt# show running-state mirror
mirror ops]east!@#$$%^&
    description mirror session named ops]east!@#$$%^&
    destination interface ethernet 0/1
!
```

- CDB Validation

```
48xt# show system internal cdb path /mirrors/mirror

key /mirrors/mirror[name=ops\]east!@#$$%^&
{
  "config": {
    "description": "mirror session named ops]east!@#$$%^&",
    "name": "ops]east!@#$$%^&"
  },
  "mirror-dest-ports": {
    "mirror-dest-port": [
      {
        "config": {
          "interface": "ethernet 0/1"
        },
        "interface": "ethernet 0/1"
      }
    ]
  },
  "name": "ops]east!@#$$%^&"
}
```

- SDB Validation

```
48xt# show system internal sdb path /mirrors/mirror

key /mirrors/mirror[name=ops\]east!@#$$%^&]
{
  "mirror-dest-ports": {
    "mirror-dest-port": [
      {
        "interface": "ethernet 0/1",
        "state": {
          "interface": "ethernet 0/1"
        }
      }
    ]
  },
  "name": "ops]east!@#$$%^&",
  "state": {
    "description": "mirror session named ops]east!@#$$%^&",
    "name": "ops]east!@#$$%^&"
  }
}
```

Supported Characters

List key values can contain most ASCII punctuation and special characters, including, but not limited to:

- @
- #
- [
-]
- =
- /
- ?
- !
- Whitespace characters

Most other ASCII punctuation characters are also supported. When these characters appear in gNMI path strings, any characters that are structurally significant to path parsing must be escaped as required by gNMI path syntax. Specifically:

- \ must be escaped as \\
-] must be escaped as \]

These escaping requirements apply only to path string representation and do not restrict the actual stored values.

The following examples illustrate supported user-defined names:

1. VRF-Customer@East
2. VRF#123
3. Customer/Branch-01
4. ACL_Users!SiteA

5. ACL/Test#DC1
6. Gold_Profile@DC1
7. Remark-Map#10
8. Traffic-Map/East
9. Policy?Branch
10. Tunnel 123456
11. Tunnel-East/Backup
12. Community[List]1

The following gNMI key values are also valid:

1. peer@host
2. peer=abc=def
3. key=val]end
4. community]name\test

Characters such as @, #, , [,], =, /, ?, !, and whitespace are accepted in key values.

Module-Specific Restrictions

Some YANG modules define additional validation rules, such as:

- Regular-expression (pattern) restrictions
- Minimum or maximum length requirements
- Feature-specific naming conventions
- Reserved keywords enforced by the module implementation
- The wildcard character (*) is reserved for gNMI path wildcard matching and is not accepted in list key values used in Set requests.
- Any additional restrictions defined by an individual YANG module, feature, or application continue to apply and are considered module-specific limitations.

Values that violate these module-specific constraints are not supported, even though the platform no longer imposes a general ASCII character restriction. Refer to the documentation for the applicable feature or YANG module for supported values and naming requirements.

For example:

Object Type	Example
VRF Name	VRF#123
ACL Name	ACL/Test#DC1
QoS Remark Map	Remark-Map#10
Traffic Map	Traffic-Map/East
Tunnel Interface	Tunnel 123456

Unsupported Values

The wildcard character (*) remains reserved for gNMI wildcard matching and is not supported in list key values. For example, values such as VRF*123, ACL*Branch, and ***SPECIAL [FW]*** are rejected when used as list key values. Other printable ASCII characters are supported unless restricted by the corresponding YANG module or feature implementation.

Considerations for gNMI Paths

Although special characters are supported in stored values, certain characters have special meaning in gNMI path strings and must be escaped when constructing a path:

Character	Escaped Form
\	\\
]	\]

For example, the key value `community]name\test` must be encoded appropriately when included in a gNMI path string. The original value is stored and returned without modification.



Note

Except for module-specific validation rules and the reserved wildcard character (*), printable ASCII characters are supported for generic string-type fields, leaf values, and list key values. Values are stored and returned without modification.

Inband Management Support

The system allows you 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 OS ONE Switching and Routing 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.

Import 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.



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
```

Security

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

Use the TLS encryption 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.

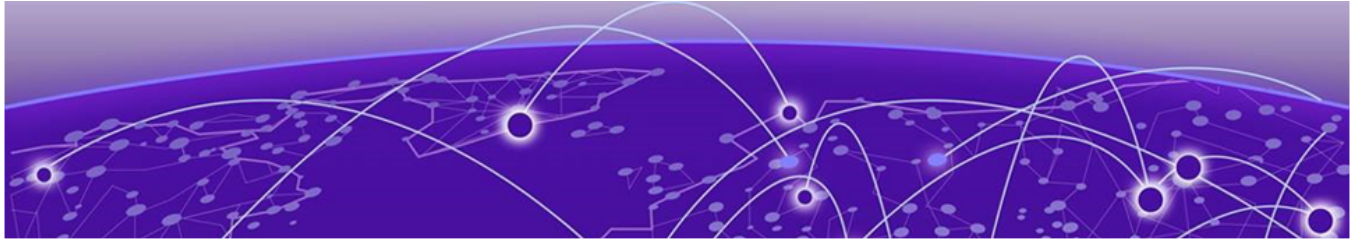
Encryption

- 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 205

[Change BMC User Password](#) on page 206

[Enable the BMC Management Interface](#) on page 206

[Configure BMC Management Interface IP Address](#) on page 208

[Reset BMC Configuration to Factory Defaults](#) on page 209

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 you 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, you need to be physically present near the device to resolve any issues.

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. OS ONE 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.

The following example shows the BMC configuration that is running currently on the device:

```
device# show running-config bmc
!
bmc
  interface 0
    no ipv4 address dhcp
    ipv4 address 198.51.100.0/24
    ipv4 gateway 192.0.2.0
    no shutdown
  !
!
```

The following example shows the BMC state that is running currently on the device:

```
device# show running-state bmc
!
bmc
  interface 0
    no ipv4 address dhcp
    ipv4 address 198.51.100.0/24
    ipv4 gateway 192.0.2.0
    no shutdown
  !
!
```

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 OS ONE configures the 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:

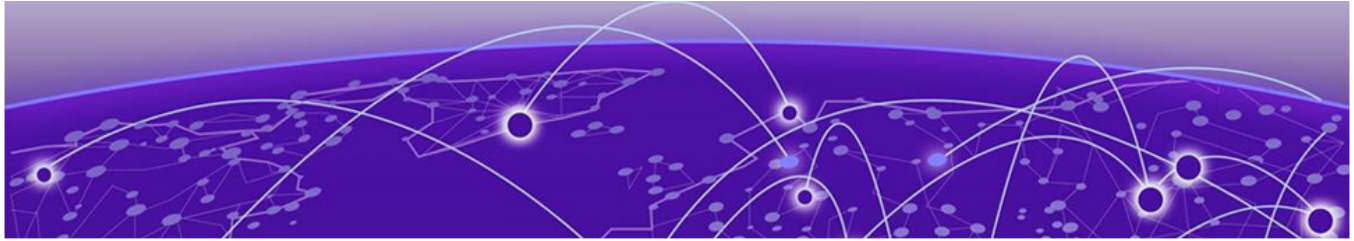
From the Executable Mode, execute the `bmc factory reset` command.

```
# 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
%Info: WARNING: BMC and System will be rebooted sequentially after bmc factory reset! Do
you want to continue? [y/n]:y
```



Port Mirroring

[How Mirroring Works](#) on page 210

[Configuration Validations for Mirroring](#) on page 211

[YANG Modules for Mirroring](#) on page 212

[Port Mirroring Configuration Using CLI](#) on page 213

[Use Case Scenarios and Configuration Examples for Port Mirroring](#) on page 214

[Remote Switched Port Analyzer \(RSPAN\) Mirroring](#) on page 216

Use this topic to learn about the configuration and implementation of port mirroring in Extreme OS ONE.

How Mirroring Works

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 source can be a physical or port channel interface; destination can be physical, internal, or port channel interfaces. Different mirroring types are configured based on the destination location. 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-based mirroring: Only Extreme 8730-32D hardware platforms support port mirroring..
 - Source: Physical/Port Channel interface
 - Destination: Physical, Port Channel, or Internal interface
 - Flow-based mirroring: ACL identifies the traffic flow to mirror.
 - Source: Physical, Port Channel, or VE interface (identified via ACL)
 - Destination: Physical, Internal, or Port Channel interface

Limitations

-
- TD3: No true egress mirroring
 - Four mirror copies per source interface; 3 available (1 used for BFDoverVxLan)
- J2: One mirror session attachment (in+out) per source interface
- "In+out" direction = 2 mirror copies
- TD4: Physical/internal interfaces only as mirror destinations
- Mirroring happens post-IFP stage; egress mirroring misses some CPU-injected packets
- Workaround: Ingress mirror on peer device for missed packets
- No flow-based egress mirroring on TD3 and TD4

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. For example: when mirroring a traffic from 40G to 10G port, the traffic exceeding 10G gets dropped.
- Port-Based Mirroring: Only physical, internal, and port-channel interfaces are supported as source.
- Max mirror sessions: J2 (4), TD3 (3)

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

```

Port Mirroring Configuration Using CLI

The system provides various CLI commands to configure mirroring functionality such as mirror destination, port-based mirroring, and flow-based mirroring.

Configure Mirror Destination

Follow this procedure to configure a mirror destination.

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>
```

Configure Port-Based Mirroring

Follow this procedure to configure port-based mirroring.

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)
```

Configure Flow-Based Mirroring

Follow this procedure to configure flow-based mirroring.

1. Enter interface configuration mode for the specified interface.

```
device(config)# interface ethernet 0/1
```

2. Create a mirror session.

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)
```

3. Specify the destination interface.

The command sets the destination interface for the mirror session.

```
device(config-mirror)# destination interface (ethernet | internal | port-channel) <NAME>
```

4. Create a mirror description.

```
device(config-mirror)# description flow based mirroring
```

5. Configure an ACL with mirroring.

```
device(config)# ipv4 access-list <ACCESS_LIST_NAME>
switch(config-ip-acl)# seq 10 permit ipv4 any any mirror <MIRROR_SESSION_NAME>
```

6. Apply ACL to an interface.

```
device(config)# int ethernet <NAME>
device(config-interface)# ipv4 access-list a1 in
device(config-interface)# ipv4 access-list a1 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

- Source: Physical, Port Channel interfaces

- Destination: Physical, Internal, Port Channel interfaces

1. Mirror session configuration 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. Mirror session configuration 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. Mirror session attachment to a physical interface

```
device(config-mirror)# int eth 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 mirror_session_5
    destination interface ethernet 0/6
!
mirror mirror_session_6
    destination interface port-channel 20
!
device#
```

5. Display the configuration of the mirror attachment under the interface.

```
device# show running-config interface ethernet 0/8
interface ethernet 0/8
    shutdown
    mirror mirror_session_6 in
    mirror mirror_session_6 out
!
device#

device# show mirror all
Number of mirrors: 2
        Name : mirror_session_5
        Description : -
        Destination Interface : ethernet 0/6

        Name : mirror_session_6
        Description : -
        Source Interface : ethernet 0/8
        Direction : MIRROR_INGRESS
        Source Interface : ethernet 0/8
        Direction : MIRROR_EGRESS
        Destination Interface : port-channel 20
device#
```

Local SPAN Flow-Based Mirroring

- Source: Physical, VE, Port Channel interfaces
- Destination: Physical, Internal, Port Channel interfaces

1. Mirror session configuration with Physical interface as Mirror destination

```
device(config)# mirror mirror_session_5
device(config-mirror)# description flow based mirroring
```

```
device(config-mirror)# destination interface ethernet 0/6
device(config-mirror)#
```

2. Mirror session configuration with Port channel interface as Mirror destination

```
device(config)# mirror mirror_session_6
device(config-mirror)# destination interface port-channel 20
device(config-mirror)#
```

3. Mirror session attachment under physical and port channel interface

```
device(config)# ipv4 access-list a1
device(config-ip-acl)# seq 10 permit ipv4 any any mirror mirror_session_5
device(config-ip-acl)#

device(config)# interface ethernet 0/30
device(config-if-eth-0/30)# ipv4 access-list a1 in
device(config-if-eth-0/30)# ipv4 access-list a1 out
device(config-if-eth-0/30)#
```

4. Running configuration of mirror sessions

```
device# show running-config mirror mirror_session_5
mirror mirror_session_5
  description flow based mirroring
  destination interface ethernet 0/6
!
device#
```

5. Running configuration of flow-based mirror attachment under interface

```
device# show running-config access-list
ipv4 access-list a1
  seq 10 permit ipv4 any any mirror mirror_session_5
!
device#
```

Remote Switched Port Analyzer (RSPAN) Mirroring

Remote Switched Port Analyzer (RSPAN) extends port mirroring across multiple switches by transporting mirrored traffic over a dedicated VLAN. RSPAN enables administrators to monitor traffic from a remote switch without physically connecting an analyzer device to the source switch.

Unlike local SPAN, which mirrors traffic on the same switch, RSPAN forwards mirrored packets through an RSPAN transport VLAN to a remote monitoring destination.

RSPAN provides the following advantages:

- Monitor traffic across multiple switches.
- Analyze network traffic from a centralized monitoring location.
- Eliminate the need to physically connect monitoring equipment to the source switch.
- Preserve original packet contents while transporting mirrored traffic through an RSPAN VLAN.

RSPAN vs. Local Mirroring

Understanding the differences between RSPAN and local (SPAN) port mirroring helps you choose the right approach for your network:

Feature	Local Mirroring (SPAN)	Remote Mirroring (RSPAN)
Scope	Single switch only	Multiple switches
Range	Limited to local ports	Enterprise-wide
Setup	Simple, few commands	More complex, multiple steps
Best For	Local troubleshooting	Security monitoring, remote analysis

How RSPAN Works

The RSPAN process consists of the following steps:

1. The source switch mirrors ingress, egress, or both directions of traffic from the configured source interface.
2. A dedicated RSPAN VLAN tag is added to the mirrored packet.
3. The mirrored packet is forwarded across the Layer 2 network using the configured RSPAN VLAN.
4. The destination switch receives the mirrored traffic and forwards it to the analyzer or monitoring interface

Supported Interfaces

Source Interfaces

The following interface types can be configured as mirror sources:

- Physical Ethernet interfaces
- Port-channel interfaces

Remote-Span Support Matrix

Platform	Source Limits	Supported Destination Interfaces	Max Unique Ingress Destination Per Source	Max Unique Egress Destination Per Source	Total MTP	Notes
Jericho-2 (J2)	NIL	Physical, Port-channel, Internal	1	1	32	Index 0 reserved
Trident-3 (TD3)	NIL	Physical, Port-channel, Internal	3	3	4	Index 0 reserved for BFDvVxLAN
Trident-4 (TD4)	NIL	Physical, Internal	4	8	8	All available



Note

Port-channel destinations are not supported on TD4 platforms because of hardware limitations.

Configuration Restrictions

The following restrictions apply to RSPAN:

- Only one remote-span interface can be configured per mirror session.
- A mirror session cannot simultaneously use local mirror destination ports and remote-span destinations.
- MLAG interfaces cannot be configured as remote-span destinations.
- VLAN ID must be within the range 1–4094.
- Remote-span is supported only on Ethernet, Internal, and Port-channel interfaces.
- Overlapping VLAN bridge-domain and remote-span configurations are not allowed.

Unsupported Operations on Remote-Span VLANs

Do not configure the following features on a remote-span bridge domain:

- IP interface attachment
- Static MAC configuration
- BD-VNI association
- ARP snooping
- ARP suppression
- Similar Layer 2 or Layer 3 service features

Validation Messages

The system generates the following errors when invalid configurations are attempted.

CLI Validations	Error Message
Maximum Session Limit Reached	Error: Maximum number of mirror sessions reached
Local Destination and RSPAN Conflict	Error: "Mirror <name>: Cannot configure remote-span when mirror-dest-ports is already configured. Mirror can use either port-based mirroring OR VLAN-based remote-span, not both Error: Mirror <name>: Cannot configure mirror-dest-ports when remote-span is already configured. Mirror can use either port-based mirroring OR VLAN-based remote-span, not both
Multiple Remote-Span Destinations	Error: Only one remote-span interface can be configured for the mirror
Invalid VLAN Range	Remote-span VLAN must be in range 1-4094
Unsupported Destination Interface Type	Error: Remote-span is supported only on ethernet, port-channel and internal interfaces
BD Already Exist	Cannot configure remote-span VLAN %d: bridge-domain %d already exists in vlan mode Cannot configure bridge-domain in vlan mode with VLAN %d: remote-span is already configured with the same VLAN
MLAG Interface Restriction	Error: MLAG interface %s is not allowed as remote-span destination.

Configure RSPAN Mirroring

Before configuring RSPAN:

- Create a dedicated VLAN for RSPAN transport.
- Ensure Layer 2 connectivity exists between source and destination switches.
- Configure the RSPAN VLAN on all intermediate switches.

- Verify that the destination analyzer interface is available.
- Ensure the configured VLAN ID is within the range of 1–4094.

**Note**

- Use a dedicated VLAN exclusively for RSPAN traffic.
- Avoid configuring normal production traffic on the RSPAN VLAN.
- Verify end-to-end VLAN propagation across all intermediate switches.
- Ensure sufficient bandwidth between source and destination switches to carry mirrored traffic without loss.

Follow this procedure to configure RSPAN on the source switch.

1. Configure RSPAN on the source switch.

a. Create an RSPAN mirror session.

```
switch(config-mirror)# remote-span
destination Remote-span destination configuration
switch(config-mirror)# remote-span destination interface ethernet 0/10 vlan 10
```

This configuration:

- Creates mirror session `rspan-session`.
 - Uses Ethernet interface 0/1 as the remote-span destination.
 - Encapsulates mirrored traffic with VLAN 10.
- b. Attach the mirror session to a source interface.

```
switch(config)# interface ethernet 0/10
switch(config-if)# mirror rspan-session in
```

You can configure:

- `in` for ingress traffic
- `out` for egress traffic

2. Configure intermediate switches.

Intermediate switches transport mirrored traffic but do not terminate the RSPAN session.

Configure the RSPAN VLAN as a remote-span bridge domain:

```
switch(config)# bridge-domain 10 mode vlan remote-span
```

Add the required member interfaces or port channels that carry the RSPAN VLAN between switches.

Remote-span VLAN behavior on intermediate switches:

- Mirrored traffic is forwarded only.
- MAC learning is disabled.
- Mirrored packets are not processed as local traffic.
- Normal bridge-domain features are restricted.

3. Verify the RSPAN configuration.

a. Verify the configured mirror session:

```
show running-config mirror
```

b. Verify interface mirror attachments:

```
show running-config interface ethernet <interface-id>
```

c. Verify the remote-span bridge-domain configuration:

```
show running-config bridge-domain
```

Example Deployment:

- Source Switch

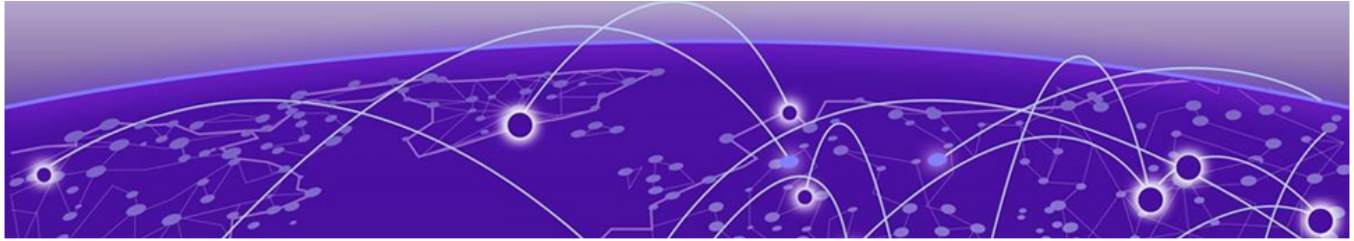
```
mirror rspan-session  
remote-span destination interface ethernet 0/1 vlan 10  
  
interface ethernet 0/10  
mirror rspan-session in
```

- Intermediate Switch

```
bridge-domain 10 mode vlan remote-span
```

- Destination Switch

Configure VLAN 10 connectivity and connect the analyzer device to the interface specified in the RSPAN mirror session. The analyzer receives mirrored packets transported through the RSPAN VLAN.



Maintenance Mode

[How Maintenance Mode Works](#) on page 222

[Enable or Disable Maintenance Mode](#) on page 222

[MLAG and BGP Module Behavior](#) on page 223

[CLI Commands for Maintenance Mode](#) on page 224

[RASLog and SNMP Traps for Maintenance Mode](#) on page 226

[Enter Maintenance Mode Before Performing Device Maintenance](#) on page 227

[Rebooting into Maintenance Mode](#) on page 228

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.

How Maintenance Mode Works

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.

Enable or Disable 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 Peer Node:** Notify the MLAG peer node 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:** Disable all front-panel ports (except internal and RME ports).
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 Peer Node:** Notify the MLAG peer node to enable secondary keepalive detection.
2. **Bring up the delay timer:** MLAG peer node 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 starts. 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.

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
Enable on reload      : disabled
Maintenance Mode      : enabled
Maintenance Mode Status : Complete
device#
```

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
    !
!
device#

```

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

```

2. **Debug Commands:** The **system internal service mlag command show-peerdb** and **system internal service mlag command show-peer args detail=1** commands provide additional information about MLAG peers and the Maintenance Mode state.

```

leaf# system internal service mlag command 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:

```
leaf# system internal service mlag command show-peer args 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 Lifs: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 the *Extreme OS ONE Switching and Routing 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)).

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 MLAG 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 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)#
```

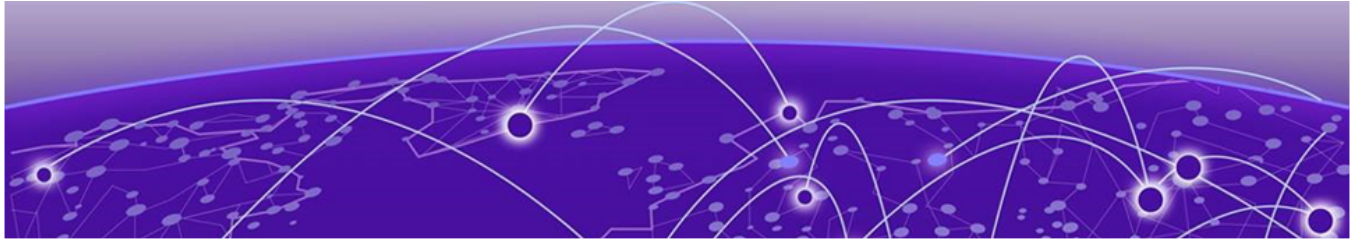
Rebooting into Maintenance Mode

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 MLAG 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 OS ONE 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)#
```



License Management

[Why OS ONE Devices Require License](#) on page 229

[Deliverables](#) on page 230

[OS ONE License Persistence](#) on page 230

[CLI Commands and Log Messages for OS ONE License Management](#) on page 230

Learn about the license management in Extreme OS ONE.

Why OS ONE Devices Require License

Extreme OS ONE devices require license keys or Self-Authenticated Upgrade (SAU) licenses to enable specific features. Each hardware platform has its own set of license types, and customers can purchase and install licenses to access licensed features.

Extreme OS ONE supports **Self-Authenticated Upgrade (SAU) Licenses**.

Key Points

- Licenses are required to enable specific features or sets of features.
- Node-Locked Licenses are not supported.
- SAU licenses provide temporary access to features without a physical license, but require EULA acceptance and license purchase within 30 days.

Self-Authenticated Upgrade (SAU) Licenses

SAU licenses enable the use of licensed features without requiring physical license installation or activation. To use SAU licenses, you must accept the End-User License Agreement (EULA) via CLI or RPC. After accepting the EULA, you can configure and use the licensed features.

Key Characteristics

- **No physical license required:** Enables feature access without installation.
- **EULA acceptance required:** You must accept the EULA before using licensed features.
- **Temporary license-free period:** Requires purchasing a license within 30 days of accepting the EULA.

Deliverables

Key deliverables for effective OS ONE license management and constraint planning include:

- The SAU (Self-Authenticated Upgrade) license or EULA (End-User License Agreement) with Advanced features will activate IAH and BGP_EVPN functionality on all platforms, except for the 8730.

On the 8730, an Advanced Core license is required to enable IAH. Note that the Advanced feature license will only enable BGP_EVPN functionality.

- Corresponding CLI and RPCs to manage ADVANCED_FEATURES and ADVANCED_CORE licenses on Extreme OS ONE.
- Licensed feature configurations are blocked until the EULA is accepted.
- IAH and BGP EVPN configurations are restricted until EULA acceptance, with specific error messages and blocked CLI and YANG paths.

OS ONE License Persistence

The following table outlines the License persistence rules, specifying when licenses are retained or removed:

Case	License retained or removed
reboot	Retained
FW upgrade	Retained
Copy default running	Retained
Fullinstall	Retained
Fullinstall no-preserve	Removed
ONIE install	Removed
Factory reset	Removed

CLI Commands and Log Messages for OS ONE License Management

Use this topic to learn about the commands and event log messages.

CLI Commands

The following commands are applicable for License Management:

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

- **show license [eula]**

Use the **show license** command to display license features, descriptions, and acceptance dates.

Use the **show license eula** command to displays the End User License Agreement (EULA) statement, regardless of acceptance status.

- **license eula (accept | decline) advanced_features**

Use this command to accept or decline the End User License Agreement (EULA) for advanced features. You can run this command only with admin privileges.

- **Accept:** Enables advanced licensed features like IAH (except on 8730 platforms) and BGP EVPN (all platforms).
- **Decline:** Disables advanced licensed features.

- **license eula (accept | decline) advanced_core**

Use this command to accept or decline the EULA for ADVANCED_CORE features on the 8730 platform, enabling or disabling their use as needed.

- **Accept:** Enables advanced core features like IAH (exclusive to the 8730 platforms)
- **Decline:** Disables advanced core features.

RPCs

The following RPC is supported:

```
syntax = "proto3";

package extreme.license;

import "extreme/types/types.proto";

option go_package = "github.extremenetworks.com/Engineering/tierra-os/proto/extreme/
license";
// Specifies the semantic release version of this proto schema.
option (extreme.types.version) = "0.1.0";

service License {

    // SetEula Accept or Decline EULA (End User License Agreement)
    rpc SetEula(SetEulaRequest) returns (SetEulaResponse);

    // Get all licenses on the device
    // Optionally set type in request to EULA_INFO, to get EULA statement.
    rpc GetLicense(GetLicenseRequest) returns (GetLicenseResponse);

}
```

Event Log Messages

The log messages provide information on license EULA acceptance, declination, and expiration, with corresponding causes, remedies, and impact.

The following Security Audit log messages are generated for license-related events:

- LogID 7045 License EULA accepted successfully.
- LogID 7046: License EULA declined.
- LogID 7047: License EULA is about to expire; purchase a valid license if not already done.

The following is an example output of log messages:

```
{
  "LogID": 7045,
  "Details": {
    "Level": "Info",
    "Msg": "License EULA accepted for %s feature.",
    "Cause": "Indicates that license EULA has been accepted successfully.",
    "Remedy": "No action is required.",
    "Impact": "None"
  }
},
{
  "LogID": 7046,
  "Details": {
    "Level": "Info",
    "Msg": "License EULA declined for %s feature.",
    "Cause": "Indicates that license EULA has been declined.",
    "Remedy": "No action is required.",
    "Impact": "None"
  }
},
{
  "LogID": 7047,
  "Details": {
    "Level": "Info",
    "Msg": "EULA acceptance period is over, please purchase a valid license.
Ignore if already purchased.",
    "Cause": "Indicates that EULA license acceptance has expired.",
    "Remedy": "Purchase license, Ignore if purchased.",
    "Impact": "None"
  }
}
```