



opengear CONSOLE MANAGER

User Guide
23.10.3





CONTENTS

Contents	2
Copyright ©	15
Document Revision History	16
Safety & FCC Statement	18
Safety Statement	18
FCC Warning Statement	18
About This User Guide	20
Installation And Connection	21
Power Connection	23
Dual Power Supply	24
SNMP Alerts for Power-related Events	25
SNMP Alert Configuration	25
Device Status LEDs	25
Connecting to the Network	28
Serial Connection	28
Reset and Erase	29
Initial Settings	30
Default Settings	31
Using the Web GUI	32
Management Console Connection via CLI	33
Accessing the Web GUI CLI Terminal	34
Change the Root Password	34
Disable a Root User	36



Change Network Settings	37
MONITOR Menu	42
System Log	43
LLDP CDP Neighbors	43
Triggered Playbooks	44
Advanced Options	45
Opengear CLI Guide	46
Getting Started with ogcli	46
Basic Syntax	48
Common Configuration Examples	54
Config Shell Guide	60
Start and End a Config Shell Session	60
Navigate in the Config Shell	61
Fields, Entities and Contexts	62
Global Context Commands	64
Entity Context Commands	64
Apply or Discard Field Changes	65
Operations	67
Supported Entities	67
Example CLI Commands	69
Configuring a Port	72
Advanced Portmanager PM Shell Guide	74
Running pmshell	74
pmshell Commands	75



Custom Control Codes for Serial Ports	77
Configure Custom Control Codes	77
Configure Control Codes for a Specified Port (CLI Examples)	78
Configure a Control Code Value for All Ports	79
Docker	80
Cron	80
Options:	80
Initial Provisioning via USB Key	82
EULA and GPL	83
ACCESS Menu	84
Local Terminal	84
Access Serial Ports	85
Quick Search	86
Access Using Web Terminal or SSH	86
Serial Port Logging	87
CONFIGURE Menu	89
Configure Serial Ports	89
Edit Serial Ports	90
Configure Single Sessions for Ports	93
Single Session Enabled	93
In the Web UI	93
In Config Shell	95
Single Session Behavior	97
Autodiscovery	99
Local Management Consoles	103
Lighthouse Enrollment	104
Manual Enrollment Using UI	106



Manual Enrollment Using the CLI	107
Playbooks	107
Create Or Edit a Playbook	108
PDUs	113
Add and Configure a PDU	114
PDU Operation	117
System Alerts	118
System Alerts - General	119
Authentication	119
Configuration	119
System Alerts - Power	120
Enable Power Supply Syslog Alerts	120
System Alerts - Networking (Connection Status)	122
Configure Signal Strength Alerts	122
Network Connections	123
Network Interfaces	123
DNS Configuration	124
Configure DNS via the Web UI	125
Configure DNS via the Command Line	126
Network Aggregates - Bonds and Bridges	126
Bridges	127
Bonds	130
Spanning Tree Protocol	134
Enable STP in a Bridge	136
Bridge With STP Enabled - UI	136
Bridge With STP Enabled - OGCLI	137
Bridge With STP Disabled - OGCLI	137
IPsec Tunnels	138
Create, Add or Edit IPsec Tunnels	138
Static Routes	142



Configure Static Routes	144
Managing Static Routes via Command Line	146
Network Resilience	147
Out Of Band Failover	148
Optional Additional Probe Address	149
Enable Out-of-Band Failover	151
DNS Queries on a Dormant Failover Interface	153
OOB Failover Types & Failover Behavior	154
User Management	156
Groups	156
Permission Changes in the Web UI	157
Understanding Access Rights	157
Understanding Serial Port Access	162
Create a New Group	165
Edit an Existing Group	167
Local Users	168
Create a New User With Password	169
Create a New User With No Password (Remote Authentication)	171
Modify An Existing User Account With Password	171
Manage SSH Authorized Keys for a User Account	172
Delete a User's Account	173
Remote Authentication	173
Configure RADIUS Authentication	174
Configure TACACS+ Authentication	176
Configure LDAP Authentication	177
Local Password Policy	179
Set Password Complexity Requirements	180
Set Password Expiration Interval	182
Password Policy Implementation Rules	183
Services	184



FIPS Compliance	186
Configure FIPS	186
Enable FIPS	186
Disable FIPS	187
Verify that FIPS is enabled	187
Considerations for using the FIPS Feature	189
Brute Force Protection	192
Configure Brute Force Protection	193
Viewing Current Bans	194
Managing Brute Force Protection via Command Line	195
HTTPS Certificate	197
Network Discovery Protocols	198
File Server	200
Enable TFTP Service	201
Update The TFTP Service Storage Location	202
Routing	203
Dynamic Routing	203
Static Routing (via the ogcli)	204
OSPF Configuration	205
Managed Configuration Items	206
NEW FIELDS in REST API & CONFIG SHELL	207
REST API	207
Config Shell	208
Interfaces, Neighbors and Networks.	209
Interfaces CONTEXT	209
Neighbors CONTEXT	211
Networks CONTEXT	211
Interaction With Configuration Files	212
Confirm OSPF Neighbours	213
Wireguard Configuration	214



Viewing a WireGuard Configuration	214
Configure WireGuard through Config Shell or REST API	215
Config Shell WireGuard CONFIGURATION	216
REST API WireGuard CONFIGURATION	217
Configurable WireGuard Fields	218
WireGuard Context Sub-objects	219
Addresses	219
Peers	220
Hooks	221
Adding a WireGuard Interface to a Firewall Zone	222
SSH	222
Unauthenticated SSH to Serial Ports	223
Enable Unauthenticated SSH	224
Enable SSH	225
Enable/Disable	225
Connecting Directly to Serial Ports	226
Feature Persist	227
Properties and Settings	227
Syslog	230
Add a New Syslog Server	230
Global Serial Port Settings	231
Global Serial Port Settings Tab - Field Definitions	231
Syslog Facility Definitions	233
Syslog Severity Definitions	234
Edit or Delete an Existing Syslog Server	235
Session Settings	235
Firewall	237
Firewall Overview	238
Example WebGUI Configuration	239



Custom Rules (firewalld “rich-rules”)	242
Useful Templates for use in the webGUI or CLI	244
Sample Rich Rules Templates	245
Firewall Management	246
Firewall Zone Settings	248
Port Forwarding	249
Manage Custom Rules	249
Firewall - Source Address Filtering	250
Firewall Source Address Bulk Services	252
Firewall Guide	254
Introduction	254
Example WebUI Configuration	255
Custom Rules (firewalld “rich-rules”)	258
Useful Templates for use in webUI or CLI	260
Sample Rich Rules Templates	261
Interzone Policies	262
Create an Interzone Policy	263
Edit or Delete an Interzone Policy	265
Customized Zone Rules	265
Adding WireGuard Zones to a Firewall	266
System	266
Administration	267
Date & Time	269
Manual Date & Time Set	270
NTP Configuration & Authentication	270
CLI Commands Associated with NTP Configuration	272
Time Zone	274
Manual Settings	274



Automatic Settings	275
Factory Reset	276
Reboot	277
Export Configuration	278
Export Configuration via Web UI	279
Export Configuration via ogcli	279
Control The Export Of Sensitive Data	280
Lighthouse Node Backup	280
Restore Configuration	281
Restore Configuration Via Web UI	281
Import Configuration via ogcli	283
System Upgrade	283
Upgrade Via Fetch From Server	285
Upgrade Via Upload	285
SNMP	285
SNMP Service	286
SNMP Alert Managers	287
Multiple SNMP Alert Managers	288
Create or Delete an SNMP Manager	288
New SNMP Alert Manager Page Definitions	289
Advanced Options	291
Config CLI GUIDE	292
Navigation in Config CLI	293
Starting a Session in Config CLI	293
Exiting a Config CLI Session	293
Navigating the Config CLI	293
Understanding Fields, Entities and Contexts	295



Global & Entity-Context Commands	298
Global Context Commands	298
Entity Context Commands	299
Config CLI Entities	300
Supported Entities	300
Config CLI Commands	308
add	309
apply	310
changes	312
delete	313
discard	315
edit	317
exit	318
help (or ?)	319
import/export	322
show	325
up / exit /	330
How Changes Are Applied or Discarded	332
Applying or Discarding Changes	333
Multi-Field Updates	335
Error Messages	339
String Values In Config Commands	340
Config CLI Use Case Examples	342
Adding a User	342
Configuring a Port	344



Configure a Single Session on a Port	346
Configure NET1 Static IPV4	348
Configure NET2 Static IPV4	348
Configure NET3 Static IPV4 for OM2224-24e units ..	348
Configure WireGuard through Config Shell	349
Root User Password - cleartext	350
Root User Password = passowrd via SHA256	350
Define Password Complexity Rules	351
Hostname	351
Contact Info	351
Time Zone and NTP	352
Create Admin User	352
Create Breakglass User (belongs to netgrp)	353
Enable netgrp - Set to ConsoleUser	353
Change SSH Delimiiter to : default is +	354
Change Port Labels	354
Enable Tacacs - Set Mode to remotelocal	355
Enable lldp on Net1 & Net2	355
Enable tftp	355
Enable Boot Messages	356
Define Session Timeouts	356
Define MOTD	356
Enable SIMM 1 Enable and Add APN	356
Enable SIMM 1 Complete End Points	357
Enable Failover	358
Add a Syslog Server	358
Set Port Logging Remote Syslog Settings	359
Enable System Monitor SNMP Traps	360
Enable SNMP V2 Service for Polling	361
Enable 2 SNMP Traps and Trap Servers	361
Create a Staic Route	362



Edit LAN (Net2) Firewall Zone	362
Edit WAN (Net1) Firewall Zone	363
Custom_rule Example for Port and Protocol	364
Enroll Into Lighthouse	364
Opengear CLI Guide	365
Getting Started with ogcli	365
Basic Syntax	367
Common Configuration Examples	373
Config Shell Guide	379
Start and End a Config Shell Session	379
Navigate in the Config Shell	380
Fields, Entities and Contexts	381
Global Context Commands	383
Entity Context Commands	383
Apply or Discard Field Changes	384
Operations	386
Supported Entities	386
Example CLI Commands	388
Configuring a Port	391
Advanced Portmanager PM Shell Guide	393
Running pmshell	393
pmshell Commands	394
Custom Control Codes for Serial Ports	396
Configure Custom Control Codes	396
Configure Control Codes for a Specified Port (CLI Examples)	397
Configure a Control Code Value for All Ports	398
Docker	399



Cron	399
Options:	399
Initial Provisioning via USB Key	401
EULA and GPL	402
UI Button Definitions	403



COPYRIGHT ©

Opengear Inc. 2024. All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Opengear. Opengear provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose.

Opengear may make improvements and/or changes in this manual or in the product (s) and/or the program(s) described in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

23.10.3	Copyright ©	15
---------	-------------	----

DOCUMENT REVISION HISTORY

Document Version Number	Revision Date	Description
22.11.0	November 2022	Updates to: Group Permissions - enhancements NTP Configuration added Serial Port logging data counters Serial Port autodiscovery System Alerts - UI layout changes
23.03	March 2023	Added CM8100-10G SKU items to Guide 10G Updates to OOB Failover Updates to OOB Failover - additional probe address added. Added Firewall - Source Address Filtering.
23.11.0	November 2023	Dual DC power supply option Support for OSPF and WireGuard New user password limitations (cannot use 'default') Firewall custom rules updated Firewall source address updated Remote syslog references removed Configure Single Session on a Port
24.2.7	February 2024	FIPS Compliance information added

		Config CLI User Guide added Config Diff tool added in CLI Guide



SAFETY & FCC STATEMENT

SAFETY STATEMENT

Please take care to follow the safety precautions below when installing and operating the Console Manager:

- Do not remove the metal covers. There are no operator serviceable components inside. Opening or removing the cover may expose you to dangerous voltage which may cause fire or electric shock. Refer all service to Opengear qualified personnel.
- To avoid electric shock the power cord protective grounding conductor must be connected through to ground.
- Always pull on the plug, not the cable, when disconnecting the power cord from the socket.

Do not connect or disconnect the appliance during an electrical storm. Also use a surge suppressor or UPS to protect the equipment from transients.

FCC WARNING STATEMENT

This device complies with Part 15 of the FCC rules. Operation of this device is subject to the following conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference that may cause undesired operation.

23.10.3	Safety & FCC Statement	18
---------	------------------------	----



Proper back-up systems and necessary safety devices should be utilized to protect against injury, death or property damage due to system failure. Such protection is the responsibility of the user.

This device is not approved for use as a life-support or medical system.

Any changes or modifications made to this device without the explicit approval or consent of Opengear will void Opengear of any liability or responsibility of injury or loss caused by any malfunction.

This equipment is for indoor use and all the communication wiring are limited to inside of the building.



ABOUT THIS USER GUIDE

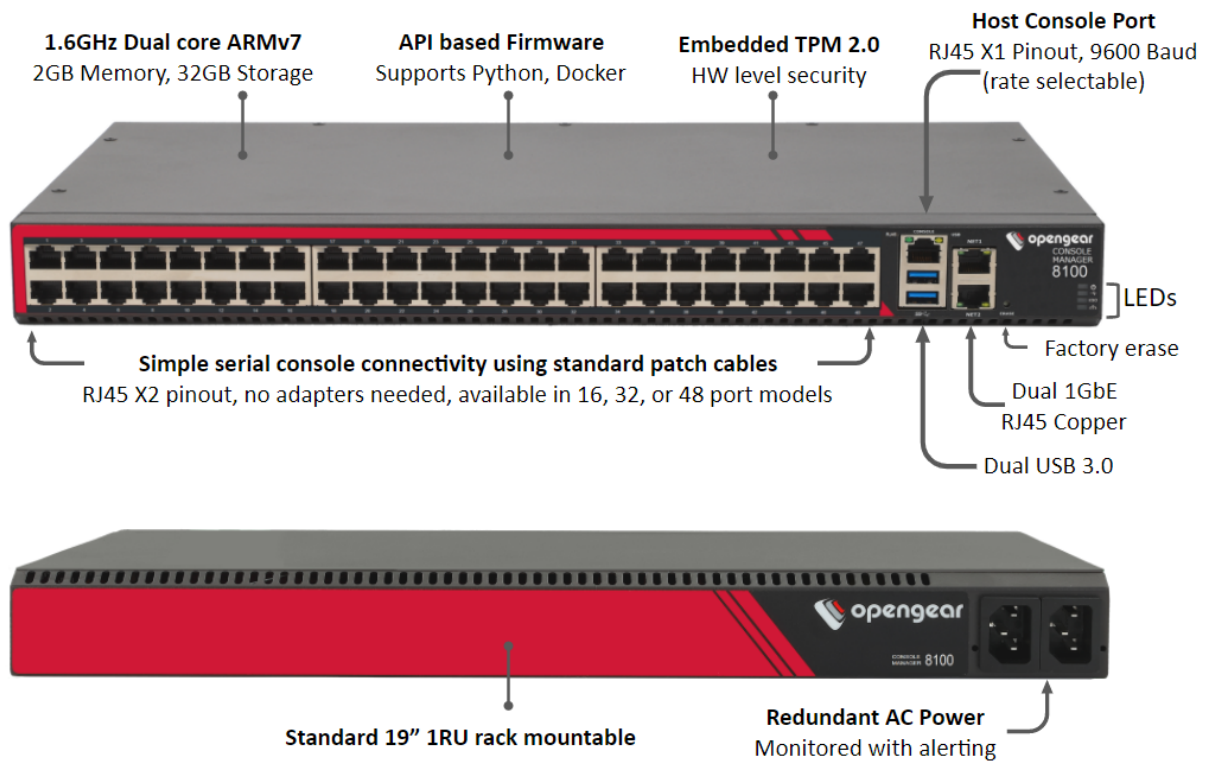
This user guide is up to date for the 23.10.3 firmware release. When using a minor release there may or may not be a specific version of the user guide for that release.

23.10.3	About This User Guide	20
---------	-----------------------	----

INSTALLATION AND CONNECTION

This section describes how to install the appliance hardware and connect it to controlled devices.

CM8100 Features:



CM8100-10G Features:

The following features apply to the CM8100-10G model:

- Static IP on Net 3.
- Pin out switching by software selectable pinout.
- Two additional 10G SFP+ fiber interfaces.

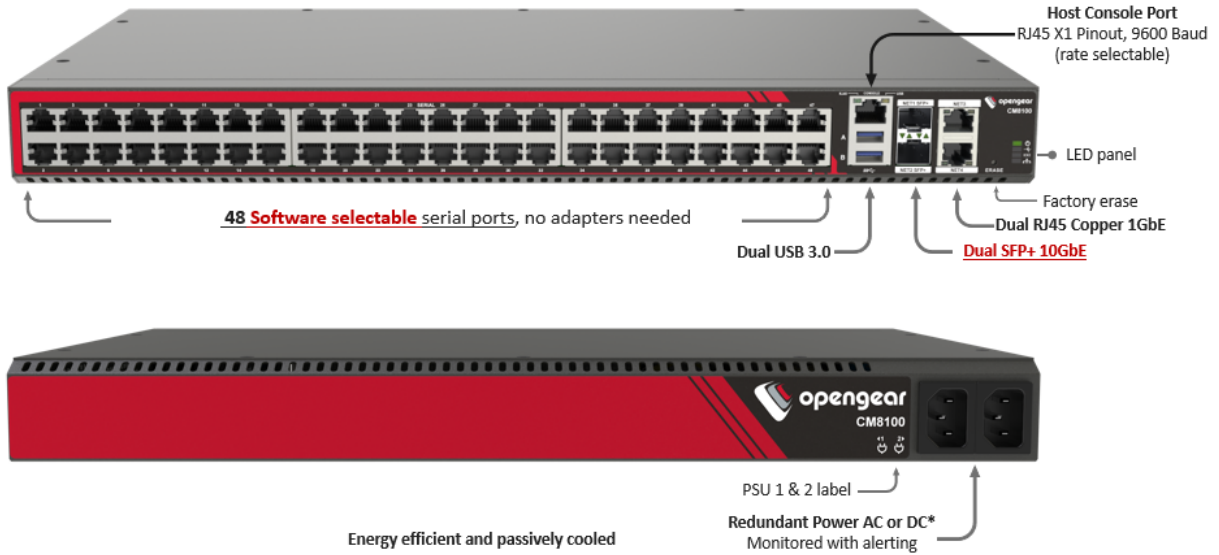
CM8148-10G

23.10.3	Installation And Connection	21
---------	-----------------------------	----

1.6GHz Dual core ARMv7
2GB Memory, 32GB Storage

API based Firmware
Supports Python, Docker

Embedded TPM 2.0
HW level security

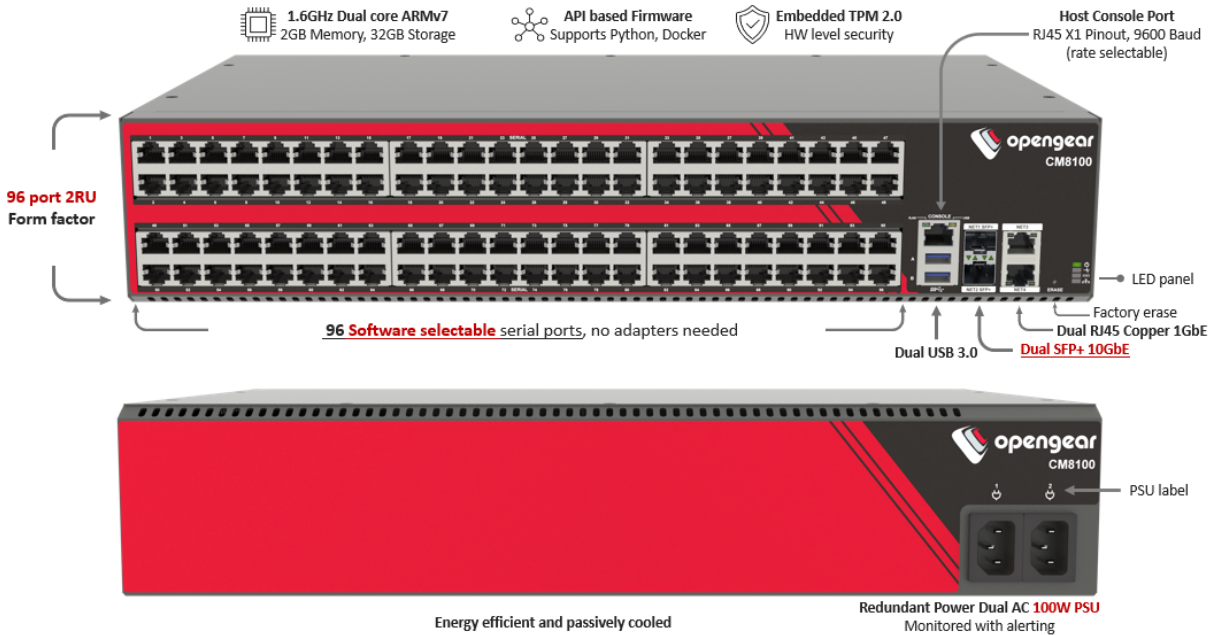


CM8196-10G

1.6GHz Dual core ARMv7
2GB Memory, 32GB Storage

API based Firmware
Supports Python, Docker

Embedded TPM 2.0
HW level security





POWER CONNECTION

The CM8100 models have dual power inlets with auto failover built in. These power supplies accept AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz. The CM8100 typically draws a maximum of 15W.

Two IEC AC power sockets, which use conventional IEC AC power cord, are located on the power side of the metal case.

Dual DC Power Supply. DDC models have a dual DC power supply with screw-in DC terminals (supplied).

Note:Country specific IEC power cords are included with the CM8100.

See also ["Dual Power Supply" on the next page](#) and ["System Alerts - Power" on page 120](#).

Console Manager Platform (CM8100) Environmental And Power	
Power Supply	Dual AC
Power Draw CM8100	Typical ly <15W
Power Draw CM8100-10G	CM8148-10G <25W Typical CM8196-10G <30W Typical.
Operating conditions	Temperature 5~50C, Rel Humidity 5~90%
Cooling	Passive

Power Draw Sensors	Active multi-zone power draw monitoring of 12V power. No monitoring on 120V AC.
--------------------	--

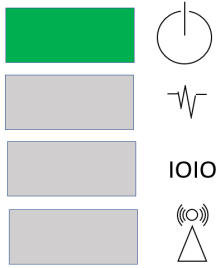
DUAL POWER SUPPLY

Dual Power Supply, including Dual DC (DDC) can provide power redundancy for devices, especially those that may operate in harsher environments. A secondary power supply provides redundancy for the device if one PSU is unplugged or in the event of a failure.

LED POWER STATUS INDICATOR

The power LED indicator requires no configuration and will display the dual power status on the Console Manager device.

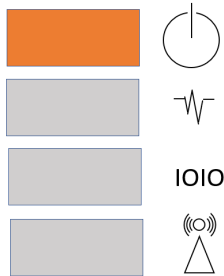
On a **dual** PSU device that has power connected to *two* PSUs, the LED power status indicator should be green at all times.



The legend shows four rows of colored squares with corresponding icons to their right. The first row has a green square and a power button icon. The second row has a grey square and a lightning bolt icon. The third row has a grey square and the text 'IOIO'. The fourth row has a grey square and a wireless signal icon.

If a **dual** PSU device has power connected to *one* PSU (power supply unit), the LED power status indicator is colored amber indicating that the unit has no

redundancy in the event of a power failure.



SNMP ALERTS FOR POWER-RELATED EVENTS

The System Voltage Range SNMP alert is triggered when there is a change in power status such as a system reboot or when the voltage on either power supply leaves or enters the configured range of the System Voltage alert.

SNMP ALERT CONFIGURATION

The System Voltage Range SNMP alert is configured in the Configure > SNMP Alerts page, see ["System Alerts - Power" on page 120](#).

DEVICE STATUS LEDS



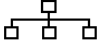
The LED states shown below are determined through infod status and config-server data. The config server holds a configurable threshold value for the Cell LED Amber / Green light, and modem enabled / disabled information.

Status LEDs

23.10.3

Installation And Connection

25

LED Condition					
	LED Off	Amber Flashing	Amber Solid	Green Flashing	Green Solid
Power 	Device is off.		On a dual power supply system: Only one PSU is connected.		On a single power supply system: power is connected. On a dual power supply system: Redundant power is connected.
Heartbeat 	Device has halted.	Device is booting.		Normal operation.	Device is halted.
Network 	No active network connection	Device is failover starting.	Device is in failover.	Normal network connection is stopping or normal network is up and failover is stopping.	Network is connected.
Status LEDs (continued).					
LED Condition					

	LED Off	Amber Flashing	Amber Solid	Green Flashing	Green Solid
CM8100 ONLY					
NET1	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G
NET2	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G
CM8100-10G ONLY					
NET1	No active network connection	Network activity	Network link	Network activity 10G	Network link 10G
NET2	No active network connection	Network activity	Network link	Network activity 10G	Network link 10G
NET3	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G
NET4	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G
IOIO 				Any serial activity is received, on either con-	

				sole/usb console or device serial ports.	
--	--	--	--	--	--

Note: The amber LED signal threshold config is set to 50% of normal signal strength.

For information on the setting of network and power alert thresholds, see:

["System Alerts - Networking \(Connection Status\)" on page 122](#)

["System Alerts - Power" on page 120](#)

CONNECTING TO THE NETWORK

Generally, Console Manager products have two network connections labeled NET1 and NET2. In the CM8100 there are options for copper wiring (on a standard RJ-45 connector). The CM8100-10G also has a static IP port on NET3.

The network connections on the CM8100 are located on the serial port side of the unit. Connect the provided shielded CAT5 cable to the NET1 to a computer or into your network for initial configuration. By default NET1 and NET2 are enabled.

SERIAL CONNECTION

CM8100

Serial Ports:

The serial connections feature RS-232 with Cisco Straight X2 pinout, 50 to

23.10.3	Installation And Connection	28
---------	-----------------------------	----



230,400bps. Connect serial devices with the appropriate STP cables.

Note: The CM8100-10G also offers a software-selectable pin out (Port PinOut) on all serial ports.

Console Port:

1 x RJ45 RS-232 Console Port - Cisco rolled X1 pinout, baud rate 9600

RESET AND ERASE

CONFIGURE > System > Reboot

The Console Manager reboots with all settings (e.g. the assigned network IP address) preserved.

To reboot the unit:

Select **CONFIGURE > System > Reboot**.

To erase the unit:

Push the **Erase** button on the port-side panel twice with a bent paper clip while the unit is powered on.

This resets the appliance to its factory default settings. Any modified configuration information is erased. You will be prompted to log in and must enter the default administration username and administration password (Username: root Password: default). You will be required to change this password during the first log in.

INITIAL SETTINGS

This section provides step-by-step instructions for the initial settings on your Console Manager.

By default, all interfaces are enabled. The unit can be managed via Web GUI or by command line interface (CLI).

- ["Default Settings" on the next page](#)
- ["Management Console Connection via CLI" on page 33](#)
- ["Change the Root Password" on page 34](#)
- ["Disable a Root User" on page 36](#)
- ["Change Network Settings" on page 37](#)
- For Configure Serial Ports (see ["Configure Serial Ports" on page 89](#))

Tip: There is also a Quick Start Guide to assist with easy setup of the Console Manager. The QSG is available at:
<https://opengear.com/support/documentation/>

23.10.3	Initial Settings	30
---------	------------------	----

DEFAULT SETTINGS

Tip: See also the Quick Start Guide available at the Opengear documentation web page: <https://opengear.com/support/documentation/>

The CM8100 is configured with a default static IP Address for NET1 of 192.168.0.1 Subnet Mask 255.255.255.0.

The CM8100-10G is configured with a default static IP Address for NET3 of 192.168.0.1 Subnet Mask 255.255.255.0.

SERIAL PORT SETTINGS

The default settings for the serial ports 1 up to 48 on a new device are:

“Console server” mode, 9600, 8N1, X2 (Cisco straight) pinout; the escape character is “~” .

BROWSER WEB GUI

The Console Manager offers a Web GUI via web browser that supports HTML5.

1. Type `https://192.168.0.1` in the address bar. HTTPS is enabled by default.
2. Enter the default username and password

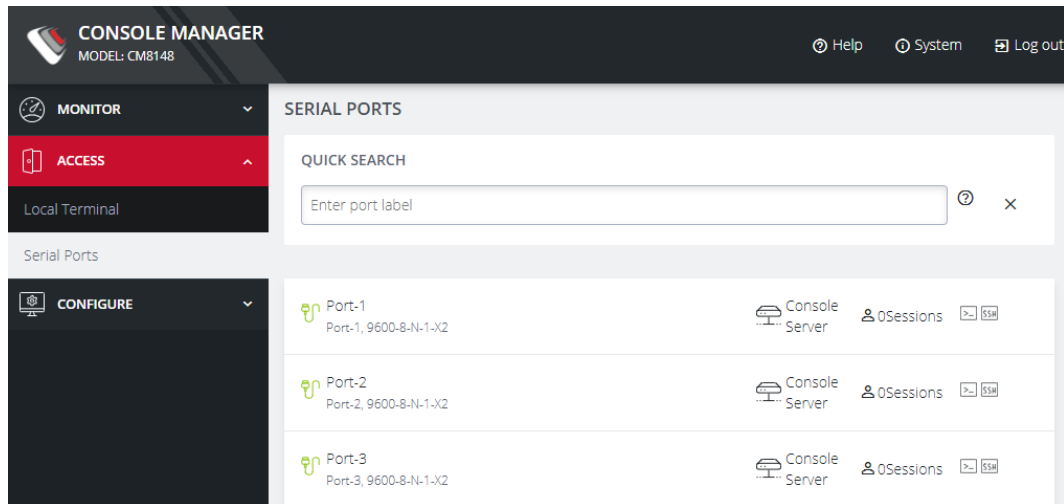
Username: root

Password: default

3. After the first successful log-in you are required to change the root password.

23.10.3	Initial Settings	31
---------	------------------	----

4. After log-in the Web GUI is available. Check system details in the top right-hand side of the Web GUI.
5. In the Navigation Bar on the left side, navigate to the **ACCESS > Serial Ports** page. The Serial Ports page displays a list of all the serial devices, including the links to a Web Terminal or SSH connection for each.



USING THE WEB GUI

The Web GUI can be switched between **Light** or **Dark** mode by adjusting the toggle on the bottom left.

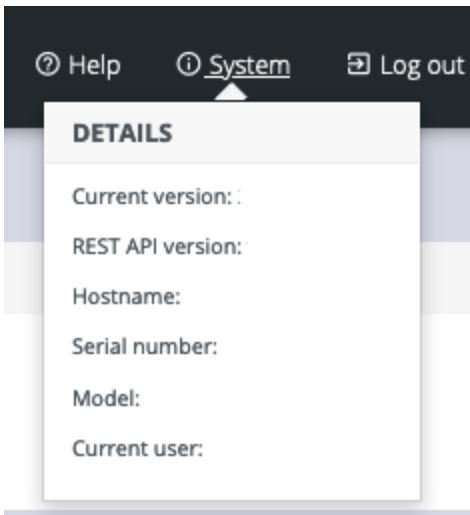


Light mode changes the user interface to display mostly light colors. This is the default UI setting. Dark mode changes the user interface to display mostly dark colors, reducing the light emitted by device screens.

The Web GUI has three menu options on the upper right: **Help**, **System**, and **Log out**.

The **Help** menu contains a link to generate a **Technical Support Report** that can be used by Opengear Support for troubleshooting. It also contains a link to the latest Console Manager User Guide.

The System menu presents the **Current version, REST API version, Hostname, Serial Number, Model, and Current user.**



MANAGEMENT CONSOLE CONNECTION VIA CLI

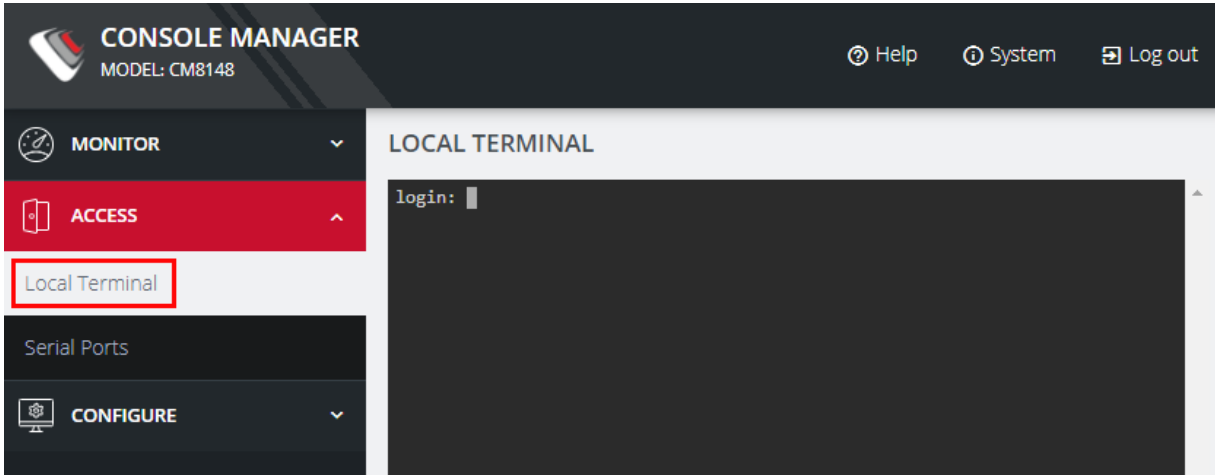
The Command Line Interface (CLI) is accessible using your preferred application to establish an SSH session. Open a CLI terminal on your desktop, then:

1. Input the default IP Address of 192.168.0.1. SSH port 22 is enabled by default.
2. When prompted, enter the log in and password in the CLI.
3. After a successful log in, you'll see a command prompt.

23.10.3	Initial Settings	33
---------	------------------	----

ACCESSING THE WEB GUI CLI TERMINAL

An alternative CLI terminal is provided within the Web GUI. To access this terminal, in the left-hand side **Navigation Bar**, navigate to the **ACCESS > Local Terminal** page. You will be required to submit your log-in credentials.



CHANGE THE ROOT PASSWORD

[CONFIGURE > User Management > Local Users > Edit User](#)

For security reasons, only the root user can initially log in to the appliance. Upon initial login the default password must be changed.

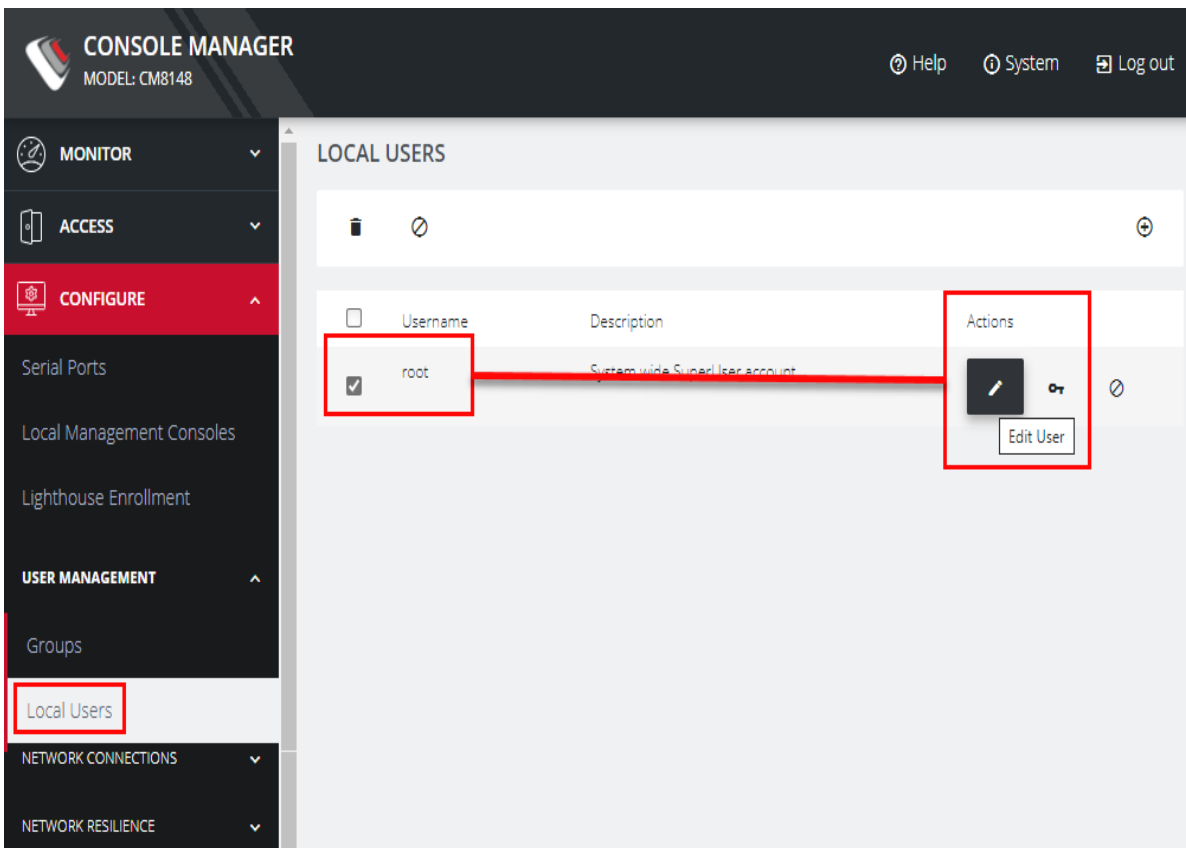
Note: Users are prevented from reusing the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This policy is applied through the WebUI, Config Shell and CLI.

Tip: Other Users' passwords may be changed using the same procedure by selecting the User's account name under the **Username** heading.

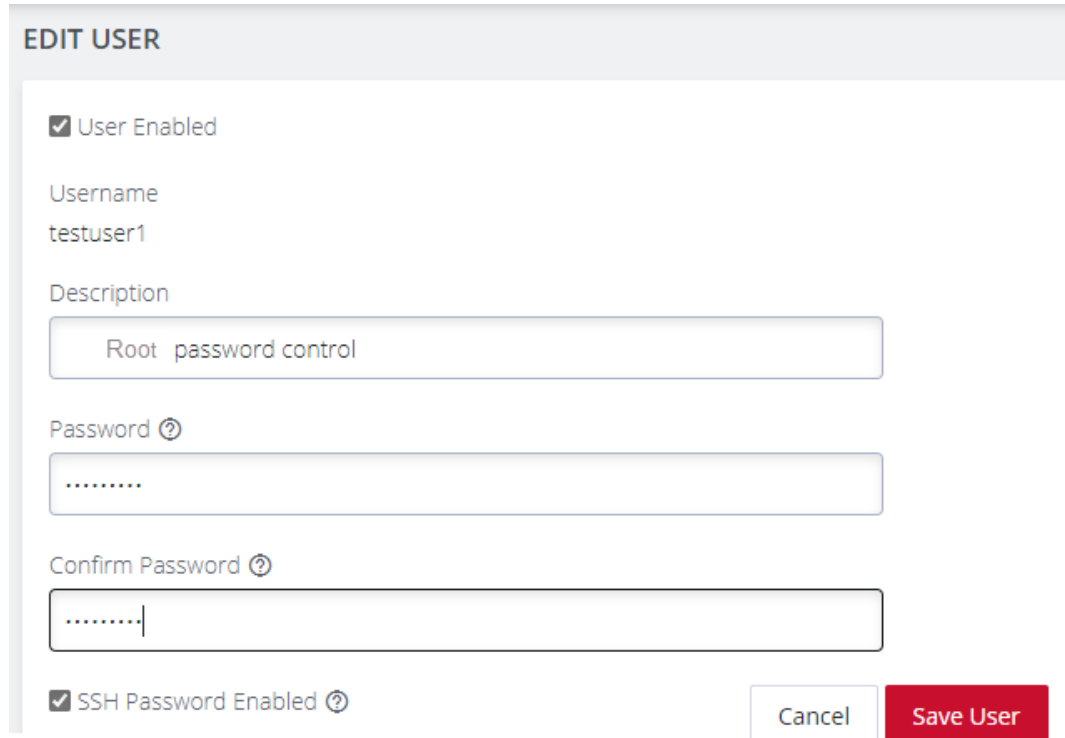
Note: The password must comply with your company's password complexity policy. See "[Local Password Policy](#)" on page 179

To change the password at any time:

1. Navigate to **CONFIGURE > User Management > Local Users**
2. Click the Root user's **Edit User** icon below the **Actions** heading.



3. In the **Edit User** page, if required, enter an optional description in the **Description** field. Enter a new password in the **Password** field and re-enter the password in the **Confirm Password** field.



EDIT USER

User Enabled

Username
testuser1

Description
Root password control

Password ⓘ
.....

Confirm Password ⓘ
.....

SSH Password Enabled ⓘ

Cancel Save User

4. Click **Save User**. A green banner confirms the password change has been saved.

DISABLE A ROOT USER

[CONFIGURE](#) > [User management](#) > [Local Users](#)

To disable a root user:

23.10.3	Initial Settings	36
---------	------------------	----

Note: Before proceeding, make sure that another user exists that has the Administrator role or is in a group with the Administrator role. For information on creating, editing, and deleting users, see "[Local Users](#)" on page 168

1. Navigate to **CONFIGURE > User management > Local Users**
2. Click the **Disable User** button in the **Actions** section next to the root user.
3. Click **Yes** in the **Confirmation** dialog.

To enable root user, log in with another user that has the Administrator role and click the **Enable User** button in the **Actions** section next to the root user.

.

CHANGE NETWORK SETTINGS

CONFIGURE > Network Connections > Network Interfaces

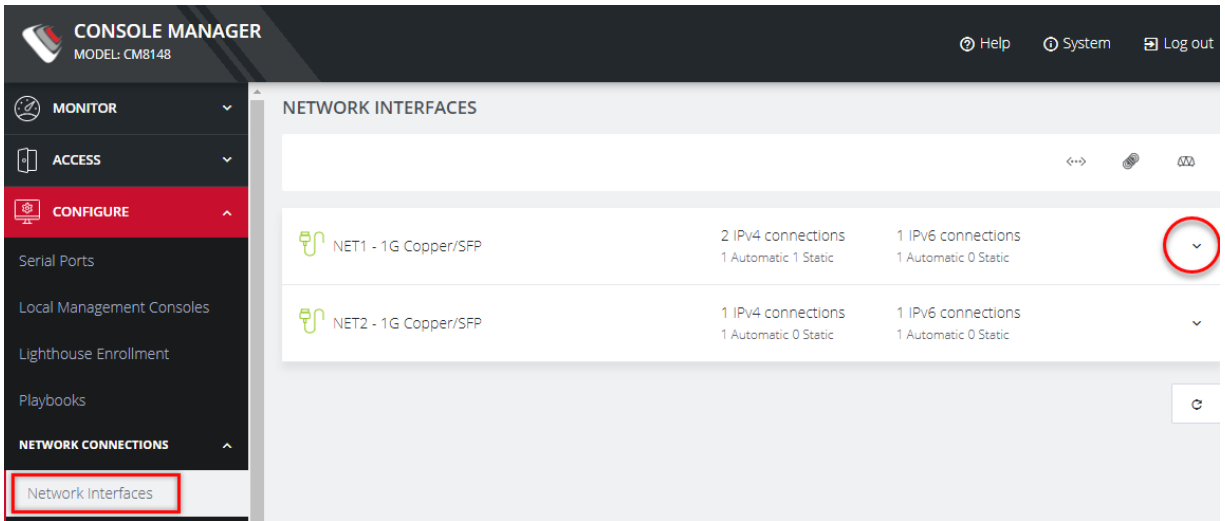
The interface supports both IPv4 and IPv6 networks. The IP address of the unit can be setup for Static or DHCP. The following settings can be configured for network ports:

- IPv4, IPv6
- Static and/or DHCP
- Enabling or disabling network interfaces
- Ethernet Media types

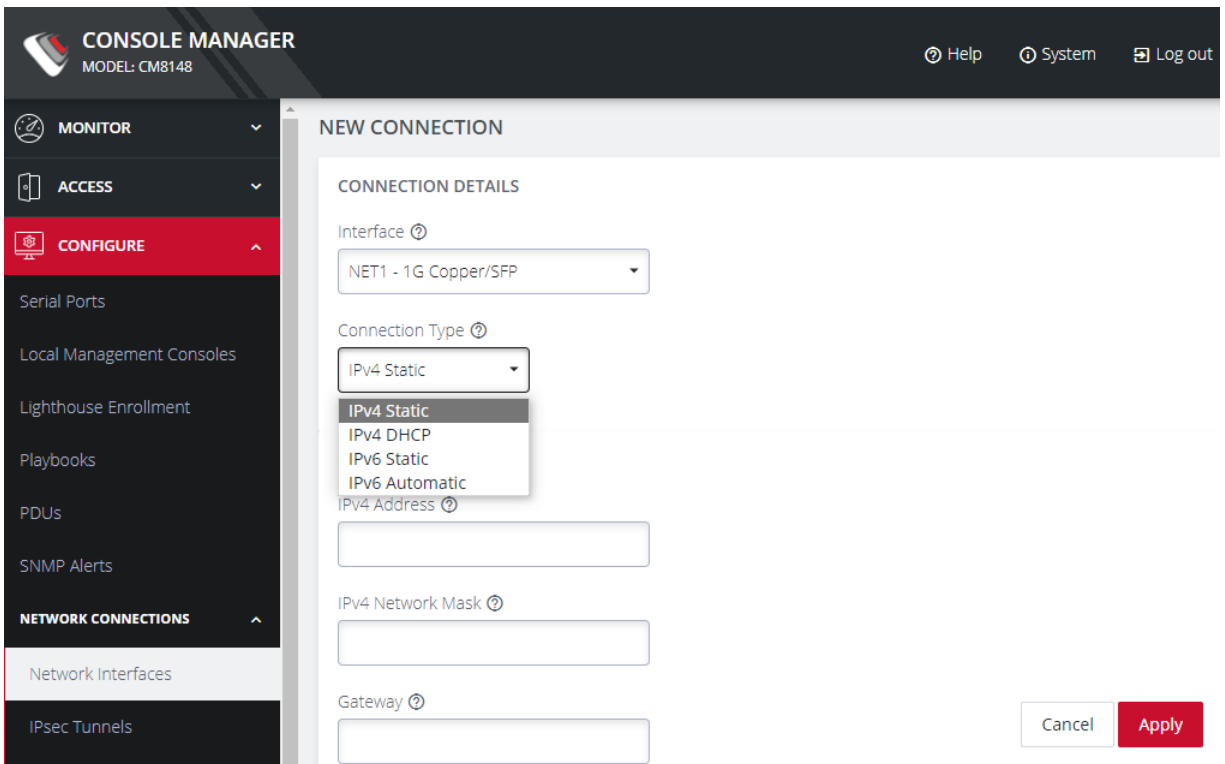
To add a new connection:

1. Click **CONFIGURE > Network Connections > Network Interfaces**

23.10.3	Initial Settings	37
---------	------------------	----



2. Click the **expand arrow** to the right of the desired interface to view its details.
3. Click the **plus icon** to open the **New Connection** page.



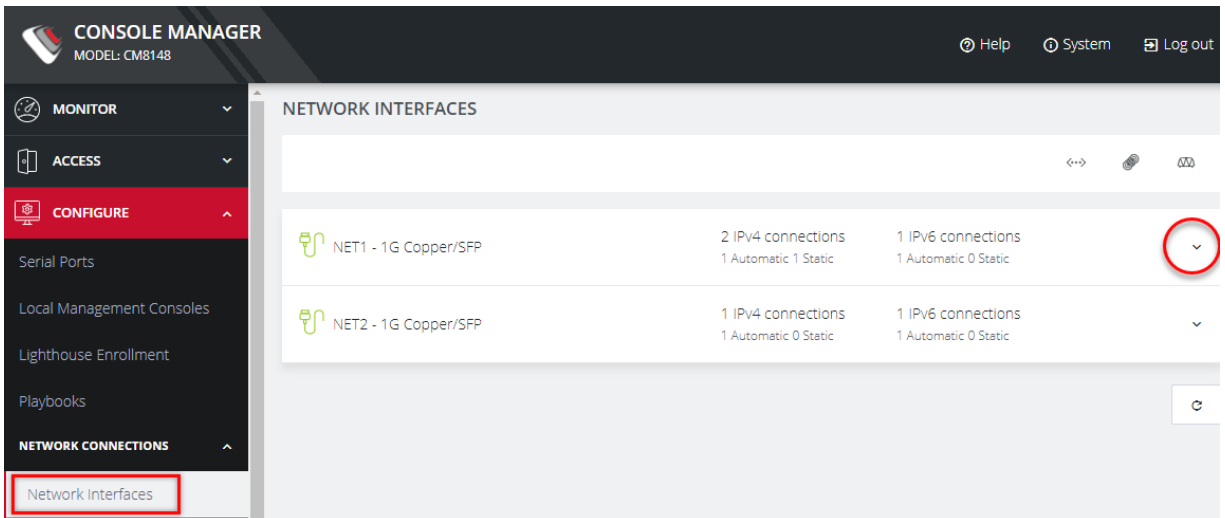
4. Select the **Interface** and **Connection Type** for your new connection.
5. The form on the bottom part of the page will change based on the **Connection Type** you choose. Enter the necessary information and click **Apply**.

To disable or delete interfaces, use the controls on the expanded section on the **CONFIGURE > Network Connections > Network Interfaces** page.

Note: If you experience packet loss or poor network performance with the default auto-negotiation setting, try changing the Ethernet Media settings on the Console Manager and the device it is connected to. In most cases, select 100 megabits, full duplex. Make sure both sides are set identically.

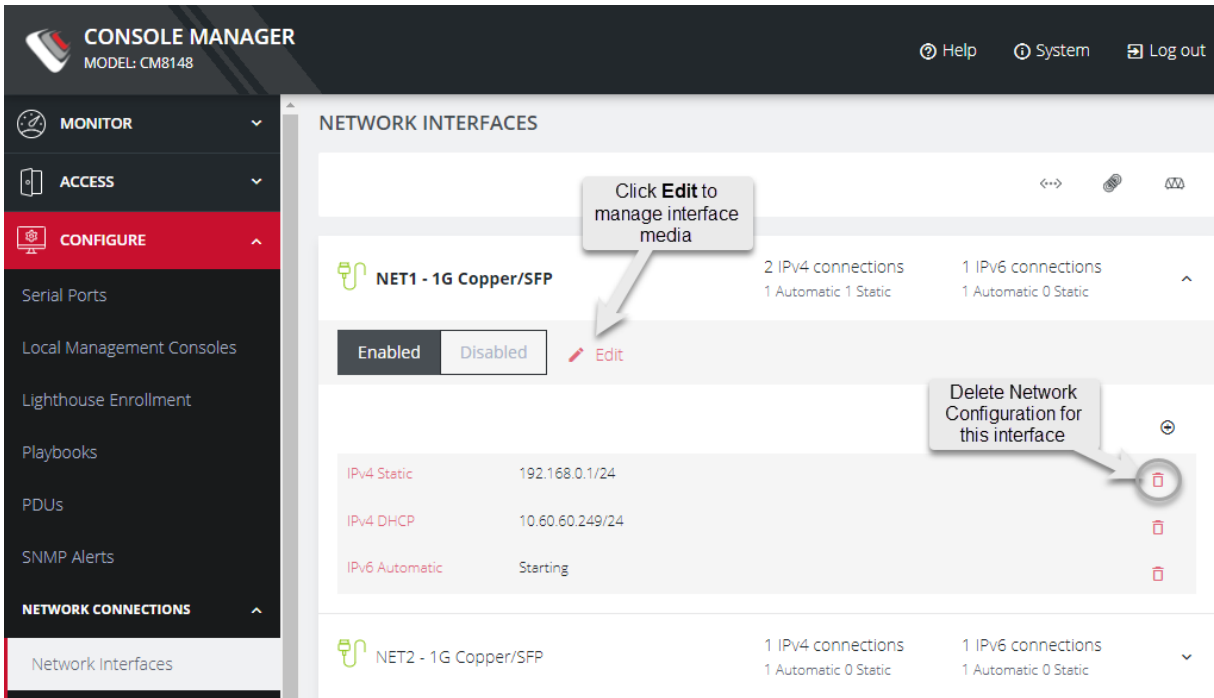
To change the Ethernet Media Type:

1. Click **CONFIGURE > Network Connections > Network Interfaces**



2. Click the expand arrow to the right of the interface you wish to modify.

23.10.3	Initial Settings	39
---------	------------------	----



3. Click **Enabled** .
4. To change the interface media setting, click the **Edit** button and edit the media settings as needed and click **Apply**.

EDIT NET1 - 1G COPPER/SFP

Interface Enabled

Media (Copper only) [?](#)

Automatic ▾

- Automatic
- 10M Half Duplex
- 10M Full Duplex
- 100M Half Duplex
- 100M Full Duplex
- 1000M Half Duplex
- 1000M Full Duplex

Name Server [?](#)

No name servers have been set

[+](#) Add Name Server

Search Domain [?](#)

No search domains have been set

[+](#) Add Search Domain

Cancel

Apply

MONITOR MENU

The MONITOR Menu is a relatively short section comprising only three topics.

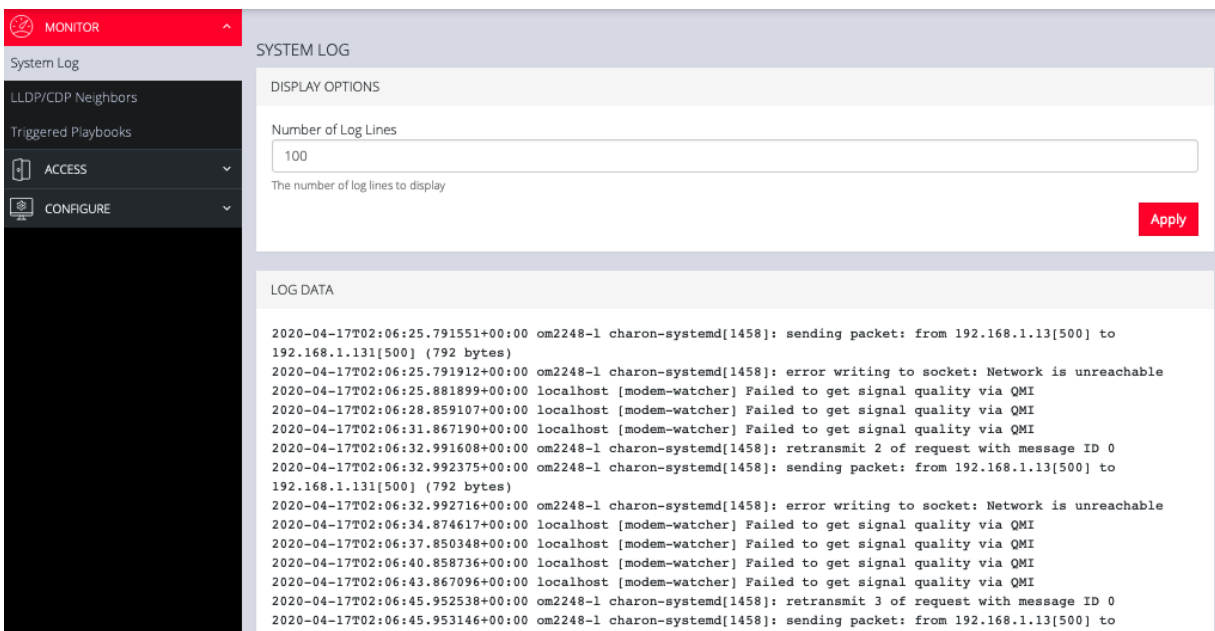
- System Log
 - Details of the system activity log, access and communications events with the server and with attached serial, network and power devices.
- LLDP/CDP Neighbors
 - Details of the LLDP/CDP Neighbors that are displayed when enabled for a connection.
- Triggered Playbooks
 - Monitoring current **Playbooks**, and applying filters to view any Playbooks that have been triggered.

SYSTEM LOG

[MONITOR > System Log](#)

The Console Manager maintains a log of system activity, access and communications events with the server and with attached serial, network and power devices.

To view the System Log, click **MONITOR > System Log**.



The screenshot shows the 'MONITOR' menu with 'System Log' selected. The 'SYSTEM LOG' page has a 'DISPLAY OPTIONS' section with a 'Number of Log Lines' input field set to '100' and an 'Apply' button. Below is the 'LOG DATA' section showing a list of system log entries with timestamps, source IP addresses, and messages such as 'sending packet', 'error writing to socket', and 'Failed to get signal quality via QMI'.

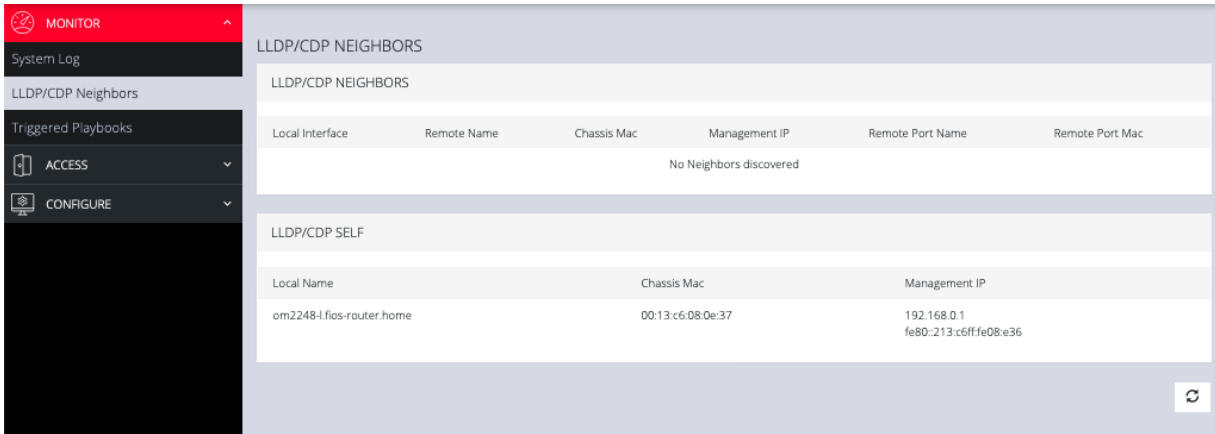
The System Log page lets you change the Number of Log Lines displayed on the screen. The newest items appear on the bottom of the list. Click the **Refresh** button on the bottom right to see the latest entries.

LLDP CDP NEIGHBORS

[MONITOR > LLDP/CDP Neighbors](#)

23.10.3	MONITOR Menu	43
---------	--------------	----

The Console Manager displays LLDP/CDP Neighbors when enabled for a connection. See **CONFIGURE > SERVICES > Network Discovery Protocols** to enable/disable.

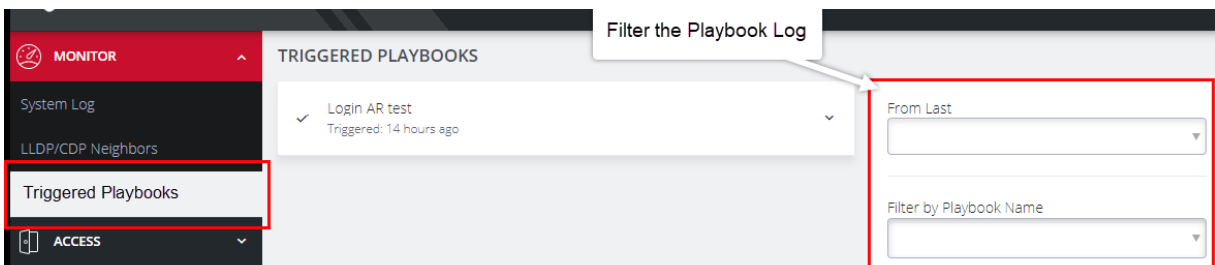


TRIGGERED PLAYBOOKS

MONITOR > Triggered Playbooks

For information on creating **Playbooks**, see the **Playbooks** topic in this User Guide.

To monitor current **Playbooks**, click on **Monitor > Triggered Playbooks**. Choose the time period if desired, and filter by **Name** of **Playlist** to view any that have been triggered.





ADVANCED OPTIONS

The Console Manager supports a number of command line interface (CLI) options and REST API.

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

external_endpoints : List of additional "address:port" endpoints to fall back to when enrolling

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

23.10.3	Advanced Options	45
---------	------------------	----



OPENGEAR CLI GUIDE

The **ogcli** command line tool is used for getting and setting configuration, and for retrieving device state and information. The purpose of ogcli is perform a single operation and exit. Operations are performed on a single entity, a list of entities, or all entities. Entities in ogcli are collections of related information items that represent device state, information or configuration.

For a list of operations supported by ogcli, see the "[ogcli Operations](#)" section.

Note:ogcli is not an interactive shell, it runs a single command and exits.

GETTING STARTED WITH OGCLI

The best way to get started with ogcli is to use the help command. Refer to the table below to access help topics within ogcli.

For detailed information about ogcli and how it works, view the ogcli help topic by running this command:

```
ogcli help ogcli
```

ACCESS OGCLI HELP AND USAGE INFORMATION

Help Command	Displays...
ogcli help	Basic ogcli help and usage information

Help Command	Displays...
ogcli help help	Detailed information about the help command
ogcli help operations	The full list of operations and a brief description of each
ogcli help entities	The full list of entities and a brief description of each
ogcli help syntax	How to get information into and out of ogcli
ogcli help ogcli	More detailed information about the ogcli tool
ogcli help usage	Common ogcli usage examples
ogcli help secrets	Detailed information about controlling the display of secrets in ogcli.
ogcli help <operation>	A description and example usage of a specific ogcli operation
ogcli help <entity>	A description of a specific entity and the operations it supports
ogcli help <entity> <operation>	An example of how to perform a specific operation on a specific entity

BASIC SYNTAX

The ogcli tool is always called with an operation, with most operations also taking one or more arguments specifying an entity for the operation to act on.

```
ogcli <operation> [argument] [argument]
```

OGCLI OPERATIONS

Operation	Description
get	Retrieve a list or single item
replace	Replace a list or single item
update	Update an item, supports partial edits
merge	Merge a provided list with existing config
create	Create an item
help	Display ogcli help
export	Export the system configuration

Operation	Description
import	Import system configuration, merging with current system configuration
restore	Import system configuration, replacing the current system configuration

SUPPLYING DATA TO OGCLI

For operations that modify an entity (e.g. 'update') the new information can be passed as inline positional arguments, but this quickly becomes cumbersome when setting a large number of fields. Information can instead be supplied through stdin by piping the contents of a file, or with Here Document (heredoc) style. The heredoc style is the most flexible format and is used extensively in ogcli examples.

HERE DOCUMENT

A here document (heredoc) is a form of input redirection that allows entering multiple lines of input to a command. The syntax of writing heredoc takes the following form:

```
ogcli [command] << 'DELIMITER'  
  
HEREDOC  
  
DELIMITER
```

- The first line starts with the ogcli command, followed by the special redirection operator << and a delimiting identifier. Any word can be used as the delimiter, commonly 'EOF' or 'END'.

- The HEREDOC block can contain multiple lines of strings, variables, commands or any other type of input. Each line can specify one field to update.
- The last line ends with the delimiting identifier used above, indicating the end of input.

```
ogcli update user <username> << 'END'  
  description="operator"  
  enabled=false  
END
```

INLINE ARGUMENTS

Field data can be entered inline with the ogcli command as arguments, with each field separated by a space.

```
ogcli update user <username> enabled=false description=\"operator\"
```

PIPES AND STANDARD INPUT

The data can also be entered via `stdin` by piping the data to the ogcli command.

```
echo 'enabled=true description="operator"' | ogcli update user  
<username>
```

Alternatively, you can provide a file via input redirection with `<`.

```
echo 'enabled=true description="operator"' > partial_record
```

```
ogcli update user <username> < partial_record
```

QUOTING STRING VALUES

All string fields require the argument to be specified with double quotes ". The shell can consume double quotes, so care must be taken when specifying strings to ensure the quotes are passed to ogcli as input.

1. Double quotes in heredoc do not need to be escaped.

```
ogcli update physif <device-identifier> << 'END'
description="test network"
END
```

2. Double quotes within single quotes do not need to be escaped.

```
ogcli update physif user <username> 'description="test user"'
```

3. Double quotes not within single quotes need to be escaped.

```
ogcli update physif user <username> description=\"test user\"
```

TAB COMPLETION

ogcli includes tab completion to assist with typing commands. When entering the start of a command, press the **<tab>** key to complete the phrase to the nearest match.

If there are multiple matches, all options will be displayed for your reference.

```
root@om1208-8e:~# ogcli get cel
cellmodem          cellmodem/sims    system/cell_reliability_test
cellfw/info        cellmodem/sims    system/cellular_logging
```

DISPLAYING SECRETS IN OGCLI

Fields containing sensitive information are called **secrets**, which are handled specially by **ogcli** to obfuscate their values when they are displayed or exported.

Passwords and private keys are examples of secret fields.

The obfuscation process provides protection against "casual observation" only and offers no cryptographic security. The **obfusc** tool can be used to obtain the clear text version of any obfuscated secret generated by any Console Manager.

For more information, view the secrets help topic by running:

```
ogcli help secrets
```

The default behavior is for secrets to be passed to ogcli in clear text, and exported or displayed in obfuscated form.

For example, setting the password:

```
ogcli update services/snmpd auth_password=\"my secret\"
```

Retrieving the password (note, the output is abridged):

```
# ogcli get services/snmpd
auth_
password="TkcxJAAAABBSB3xoFWhPA6B7sDrzq3HwaTOAO/jsURqFa0qa7hc3TA=="
```

This behavior can be overridden to display sensitive fields in clear text, obfuscated form, or masked form using the **--secrets** option. The clear text and obfuscated forms are also accepted when supplying a sensitive field.

```
# ogcli --secrets=cleartext get snmpd
auth_password="my secret"
```

```
# ogcli --secrets=obfuscate get snmpd  
auth_password="my secret"
```

```
# ogcli --secrets=mask get snmpd  
auth_password="*****"
```

If an export is performed with the **--secrets=mask** option it is impossible to subsequently import the configuration, because the secrets have been removed.



COMMON CONFIGURATION EXAMPLES

These examples contain a variety of notations and usage patterns to help illustrate the flexibility of ogcli. The examples can be copied and pasted into the CLI.

REPLACE MESSAGE OF THE DAY (MOTD) DISPLAYED AT LOGIN

```
ogcli replace banner banner=\"updated message\"
```

RETRIEVE USER RECORD

```
ogcli get user <username>
```

UPDATE ITEM WITH FIELD WHERE VALUE IS A STRING

```
ogcli update user <username> description=\"operator\"
```

UPDATE ITEM WITH FIELD WHERE VALUE IS NOT A STRING

For example, a numeric or boolean value

```
ogcli update user <username> enabled=true
```

EXPORT SYSTEM CONFIGURATION

```
ogcli export <file_path>
```

IMPORT SYSTEM CONFIGURATION

```
ogcli import <file_path>
```



RESTORE SYSTEM CONFIGURATION

```
ogcli restore <file_path>
```

COMPARE CURRENT CONFIGURATION WITH A PREVIOUSLY EXPORTED CONFIGURATION

If the template file was generated using `ogcli export <file>`, use:

```
ogcli diff <file>
```

- Config differences will be displayed in a standard diff file format.
- If the active configuration is identical to the input config a short message will indicate they are the same.

Users can also invoke a config comparison against a template file using `ogcli-diff <file>`. When using this method, the same settings are used for the comparison.

Basic help for `ogcli diff` can be accessed with `ogcli diff -h`. Detailed help for `ogcli diff` can be accessed by `ogcli help diff`. Users must have Admin rights that can access the shell to use `ogcli diff`.

ENABLE LOCAL CONSOLE BOOT MESSAGES

```
ogcli get managementports
```

```
ogcli update managementport mgmtPorts-1 kerneldebug=true
```

CREATE NEW USER

```
ogcli create user << 'END'  
description="superuser"  
enabled=true
```

```
groups[0]="admin"  
password="test123"  
username="superuser123"  
END
```

CHANGE ROOT PASSWORD

```
ogcli update user root password=\"oursecret\"
```

CREATE NEW ADMINISTRATIVE USER

```
ogcli create user << 'END'  
username="adal"  
description="Ada Lovelace"  
enabled=true  
no_password=false  
groups[0]="groups-1"  
password="oursecret"  
END
```

MANUALLY SET DATE AND TIME

```
ogcli update system/timezone timezone=\"America/New_York\"
```

```
ogcli update system/time time=\"15:30 Mar 27, 2020\"
```

ENABLE NTP SERVICE

```
ogcli update services/ntp << 'END'  
enabled=true  
servers[0].value="0.au.pool.ntp.org"
```

```
END
```

UPDATE SYSTEM HOSTNAME

```
ogcli update hostname hostname=\"system-hostname\"
```

ADJUST SESSION TIMEOUTS

```
ogcli update system/cli_session_timeout timeout=180
```

```
ogcli update system/webui_session_timeout timeout=180
```

SETUP REMOTE AUTHENTICATION WITH TACACS+

```
ogcli update auth << 'END'  
mode="tacacs"  
tacacsAuthenticationServers[0].hostname="192.168.250.21"  
tacacsMethod="pap"  
tacacsPassword="tackey"  
END
```

SETUP REMOTE AUTHENTICATION WITH RADIUS

```
ogcli update auth << 'END'  
mode="radius"  
radiusAuthenticationServers[0].hostname="192.168.250.21"  
radiusAccountingServers[0].hostname="192.168.250.21"  
radiusPassword="radkey"  
END
```

CREATE USER GROUP WITH LIMITED ACCESS TO SERIALPORTS

```
ogcli create group << 'END'  
  description="Console Operators"  
  groupname="operators"  
  role="ConsoleUser"  
  mode="scoped"  
  ports[0]="ports-10"  
  ports[1]="ports-11"  
  ports[2]="ports-12"  
END
```

VIEW AND CONFIGURE NETWORK CONNECTIONS

```
ogcli get conns
```

```
ogcli get conn system_net_conns-1
```

```
ogcli update conn system_net_conns-1 ipv4_static_  
settings.address="\192.168.0.3\"
```

```
ogcli create conn << 'END'  
  description="2nd IPv4 Static Address Example"  
  mode="static"  
  ipv4_static_settings.address="192.168.33.33"  
  ipv4_static_settings.netmask="255.255.255.0"  
  ipv4_static_settings.gateway="192.168.33.254"  
  physif="net1"  
END
```



CONFIGURE SERIAL PORTS

```
ogcli get ports
```

```
ogcli get ports | grep label
```

```
ogcli get port ports-1
```

```
ogcli update port "port05" << 'END'  
mode="consoleServer"  
label="Router"  
pinout="X2"  
baudrate="9600"  
databits="8"  
parity="none"  
stopbits="1"  
escape_char="~"  
ip_alias[0].ipaddress="192.168.33.35/24"  
ip_alias[0].interface="net1"  
logging_level="eventsOnly"  
END
```

CONFIG SHELL GUIDE

The Config Shell feature provides an interactive and familiar environment similar to older OpenGear appliances. The result is a user-experience that feels like an Interactive CLI.

Advantages of the Config Shell are:

- Items can be created or updated without being applied immediately
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information.
- Tab complete is supported for many commands.
- Built-in help (see ["Global Context Commands" on page 64](#)).
- Has a structured, tabular view when displaying lists of data.

START AND END A CONFIG SHELL SESSION

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the maintenance console.

You can exit the Config Shell by any of the following:

- Type `exit` to end the session.
- Send an EOF (**Control+D**).
- Send an INT (**Control+C**).

Note: The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATE IN THE CONFIG SHELL

The Config Shell operates in a hierarchy of entities. Due to the variety of entities, there are several ways for you to get to a place where you can make changes.

Starting at the root, enter the entity names to descend down through lower entities. Every entity name is an operation that descends into that entity. Similarly, type the names of entities higher in the hierarchy to ascend towards the root.

IDENTIFIERS:

Singleton entity	Require only the entity name to be uniquely identified.
List/item entity	The first level is the entity name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A single entity (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: ssh/authorized_keys > userid > [key_id] .
Nested fields	The Config Shell treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

FIELDS, ENTITIES AND CONTEXTS

The config shell allows you to configure a number of fields which define settings.

The fields are grouped in entities that describe a small set of functionality. For example, there is a 'user' entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

CONTEXT WITHIN CONFIG SHELL

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. For example, in the following snippet, the 'user' context is accessed and then the 'john' sub-entity is accessed causing the context to become 'user john'. The 'show' command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show
Item names for entity user
  john matt myuser netgrp root
config(user): john
config(user john):
Entity user item john
  description
  enabled true
  no_password false
```

```
password
ssh_password_enabled true
groups (array)
config(user john):
```

GLOBAL CONTEXT COMMANDS

The following commands are available on any context:

help (or '?')	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
help <entity>	Show the help for the specific entity.
help <field>	Show the help for the specific field.
show	List the available entities and fields.
<entity>	Typing the name of an entity changes the context to focus on the named entity.
exit	Exit the command shell.

ENTITY CONTEXT COMMANDS

The following commands available on any entity context:

<field>	Show the value of a field.
<field> <value>	Change the field to the specific value.

delete	Delete the current entity. This is available when the context entity is an item in a list.
add	Append a sub entity or field to the current entity. This is only available when the context entity is a list.

APPLY OR DISCARD FIELD CHANGES

When fields and entities are changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an '*' when the `show` command is used. In addition, the `changes` command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The `show` command indicates the changed field with an '*'. The `changes` command lists the changed field.

```
config(user john): description "Admin"
config(user john): show
Entity user item john
  description Admin *
  enabled true
  no_password false
  password
  ssh_password_enabled true
  groups (array)
config(user john): changes
Entity user item john (edit)
  description Admin
config(user john):
```

23.10.3	Config Shell Guide	66
---------	--------------------	----

OPERATIONS

Once a change has been made, the following commands are available:

changes	show staged changes on all entities
apply	apply changes only on the current entity
discard	discard changes only on the current entity
apply all	apply changes on all entities
discard all	discard changes on all entities

SUPPORTED ENTITIES

The following entities are supported in phase 1 of this feature and are available in release 22.06.0:

auth	Configure remote authentication, authorization, accounting (AAA) servers.
group	Retrieve or update user group information.
ip_passthrough	Passthrough entities are for retrieving / changing IP Passthrough settings.
ip_passthrough/status	The IP Passthrough status entity provides information about what part of the IP

	<p>Passthrough connection process the device is currently at and information about the connected downstream device.</p>
local_password_policy	<p>Configure the password policy for local users. This includes expiry and complexity settings.</p>
logs/portlog_settings	<p>Check and update port log settings.</p>
managementport	<p>Used for working with local management console information.</p>
port	<p>Configure and view ports information.</p>
ports/auto_discover/schedule	<p>Manage Port Auto-Discovery scheduling.</p>
system/admin_info	<p>Retrieve or change the Console Manager appliance system's information (hostname, contact and location).</p>
system/banner	<p>Retrieve or change the Console Manager appliance system's banner text.</p>
system/cloud_connect	<p>Retrieve or change the Console Manager appliance system's cloud connect configuration.</p>
system/model_name	<p>Retrieve the Console Manager appliance's Model Name.</p>
system/serial_number	<p>Retrieve the Console Manager appliance's Serial Number.</p>

system/session_timeout	Retrieve or change the Console Manager appliance session timeouts.
system/ssh_port	The SSH port used in Direct SSH links.
system/time	Retrieve and update the Console Manager's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the Console Manager's most recent firmware and REST API version.
user	Retrieve and update user information.

EXAMPLE CLI COMMANDS

ADDING A USER

In this example below, some commentary is added. Commentary added later is denoted with a `//` prefix.

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
// Move to the user entity
```

```
config: user
config(user): help add
- Add a new item for entity user.
```

The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:

Retrieve and update information for a specific user.

```
// Create the new user
```

```
config(user): add matt
config(user matt): show
Entity user item matt
  description
  enabled true
  no_password false
  password (required)
  ssh_password_enabled true
  username matt
  groups (array)
```

```
// Fill out some fields
```

```
config(user matt): password secretpassword
config(user matt): description Admin
config(user matt): show
Entity user item matt
  description Admin *
  enabled true
  password secretpassword *
```

```
ssh_password_enabled true
username matt
groups (array)
```

```
// Edit the groups
config(user matt): groups
config(user matt groups): show
Entity user item matt field groups
config(user matt groups): add // Tab completion to show available
values
admin myuser netgrp
config(user matt groups): add admin
config(user matt groups): up // Exit the groups list
```

```
// Show and apply
config(user matt): show
Entity user item matt
description Admin *
enabled true
password secretpassword *
ssh_password_enabled true
username matt
groups (array)
0 admin *
config(user matt): apply
Creating entity user item matt.
config(user matt):
```

CONFIGURING A PORT

```

config: port
config(port): help
You are here: entity port
Description for the entity:
  Configuring and viewing ports information

Names (type <name> or help <name>)
=====
USB-A  USB-E  USB-front-lower  port03  port07  port11  port15  port19  port23
USB-B  USB-F  USB-front-upper  port04  port08  port12  port16  port20  port24
USB-C  USB-G  port01           port05  port09  port13  port17  port21
USB-D  USB-H  port02           port06  port10  port14  port18  port22

Commands (type help <command>)
=====
exit  help  show  up
config(port): port01
config(port port01): baudrate // tab completion
110   1200   150   19200  230400  300   4800   57600  75
115200 134   1800   200   2400   38400  50    600   9600
config(port port01): baudrate 57600
config(port port01): label Router
config(port port01): control_code
config(port port01 control_code): break a
config(port port01 control_code): up
config(port port01): show
Entity port item port01
  baudrate      57600      *
  databits      8
  escape_char   ~
  label         Router    *
  logging_level disabled
  mode          consoleServer
  parity        none
  pinout        X2
  stopbits      1
  control_code (object)
    break      a *
    chooser
    pnhelp
    portlog
    power
    quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):

```

ADVANCED PORTMANAGER PM SHELL GUIDE

The Portmanager program allows you to access any serial port on the console server using `pmsHELL` commands. It

- Routes network connection to serial ports
- Checks permissions
- Monitors and logs all the data flowing to/from the ports
- Allows you to run power commands if the serial port is associated with a PDU outlet.

RUNNING PMSHELL

`pmsHELL` provides an environment that allows you to access and interact with serial ports via a number of command sequences. It lets you navigate between ports using the chooser command (`~m`). For example, you can use `pmsHELL` to connect to port 8 via the portmanager via the following command line sequence.

```
# pmsHELL -l port08
```

PMSHELL COMMANDS

When running `pmshell` there are a number of command sequences that you can use that begin with the `~` key.

Note:Note: If you are connected to `pmshell` via SSH, you must add an additional `~` escape sequence.

Options	Name	Result
<code>~c</code>		The Single Session feature can be enabled or disabled by editing the <code>single_session</code> field in a given port. When a user port level administration access is logged in via <code>pmshell</code> , the port configuration menu can be accessed via any port by pressing the escape character (<code>~</code> by default) followed by <code>c</code> (<code>~c</code>).
<code>~b</code>	<code>break</code>	Generates a BREAK on the serial port (if you're doing this over ssh, you'll need to type " <code>~~b</code> ").
<code>~h</code>	<code>portlog</code>	Generates a history on the serial port. Displays the traffic logs for the port - must have port logging enabled.
<code>~.</code>	<code>quit</code>	Quits <code>pmshell</code> .
<code>~p</code>	<code>power</code>	Opens the power menu for the port. The port

Options	Name	Result
		must be configured for a PDU
~u		Opens the list of user sessions, select by number to disconnect.
~m	chooser	Connects to the port menu - go back to the serial port selection menu.
~?	pmhelp	Displays help message.

CUSTOM CONTROL CODES FOR SERIAL PORTS

Custom control codes can be defined for ease of use per port or can be applied to all ports. For example, users could define a different Power Menu control code for every port, while having a single control code for View History that applies to all ports.

Custom control codes can be used by any user with access to the serial port. In order to run the shortcuts, the user presses the CTRL key + the keycode.

Note: Only Admin users can specify short-cut control codes.

CONFIGURE CUSTOM CONTROL CODES

Admin users can configure control codes for any of the `pmshell` commands through the REST API, `ogcli` and the new interactive config shell.

Control code limitations are as follows:

- Cannot set multiple control codes for a port to use the same keycode
- The available key codes are a-z, excluding 'i' and 'm' as these can be triggered by commonly used keys TAB and BACKSPACE.

To disable a certain control code for an individual port, set the port's control code to an empty string.

CONFIGURE CONTROL CODES FOR A SPECIFIED PORT (CLI EXAMPLES)

Control Codes Action	CLI Examples
<p>Set control codes for a given port. In this example, the user sets multiple control codes for port 2</p>	<pre>ogcli update port port02 << 'END' control_code.break="b" control_code.chooser="c" control_code.pmhelp="h" control_code.portlog="l" control_code.power="p" control_code.quit="q" END</pre>
<p>Clear all control codes for a given port, in this example, port 2</p>	<pre>ogcli update port port02 << 'END' control_code.break="" control_code.chooser="" control_code.pmhelp="" control_code.portlog="" control_code.power="" control_code.quit="" END</pre>

CONFIGURE A CONTROL CODE VALUE FOR ALL PORTS

To set a particular control code to one value across all serial ports, admin users can use the script `set-serial-control-codes` from the CLI as follows:

```
set-serial-control-codes CONTROL_CODE KEY
```

where:

- **CONTROL_CODE** - Must be one of the following values: `break`, `chooser`, `pmhelp`, `portlog`, `power` or `quit`.
- **KEY** - Must be a single lower case letter a-z excluding 'i' and 'm' or an empty string designated by " " which is used to clear the control code.

CONTROL CODES FOR ALL PORTS VIA CLI (EXAMPLES)

Control Codes Action	CLI Examples
Set chooser control code to CTRL-a on all ports	<pre>set-serial-control-codes chooser a</pre>
Clear chooser control code on all ports	<pre>set-serial-control-codes chooser ' '</pre>

DOCKER

Docker is a tool designed to make it easier to create, deploy, and run applications by distributing them in containers. Developers can use containers to package up an application with all of the parts it needs, like libraries and dependencies, and then ship it out as one package. Docker is running by default on the Console Manager. You can access commands by typing `docker` in the Local Terminal or SSH.

For more information on Docker, enter `docker --help`.

CRON

Cron service can be used for scheduled cron jobs runs. Daemon can be managed via the `/etc/init.d/crond` interface, and cron tables managed via `crontab`. `Crontab` supports:

Usage:

`crontab [options] file`

`crontab [options]`

`crontab -n [hostname]`

OPTIONS:

`-u <user>` define user

`-e` edit user's crontab

`-l` list user's crontab

`-r` delete user's crontab

23.10.3	Custom Control Codes for Serial Ports	80
---------	---------------------------------------	----

- i prompt before deleting
- n <host> set host in cluster to run users' crontabs
- c get host in cluster to run users' crontabs
- x <mask> enable debugging

To perform start/stop/restart on crond service:

```
/etc/init.d/crond start
```

Cron doesn't need to be restarted when crontab file is modified, it examines the modification time on all crontabs and reload those which have changed.

To verify the current crond status:

```
/etc/init.d/crond status
```

To check current cron jobs running with the following command to list all crontabs:

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

This opens a personal cron configuration file. Each line can be defined as one command to run. The following format is used:

minute hour day-of-month month day-of-week command

For example, append the following entry to run a script every day at 3 am:

```
0 3 * * * /etc/config/backup.sh
```

Save and close the file.



INITIAL PROVISIONING VIA USB KEY

Also known as “ZTP over USB”, this feature allows provisioning an unconfigured (factory erased) unit from a USB storage device like a thumb drive.

The USB device must contain a filesystem recognized by the CM (currently FAT32 or ext4) with a file named manifest.og in the root directory. This file specifies which provisioning steps will be done. An article with a partial description of the file format is here:

<https://opengear.zendesk.com/hc/en-us/articles/115002786366-Automated-enrollment-using-USB>

The USB device can be inserted any time (before or after power is applied to the unit) and as long as the unit is unconfigured, the ZTP over USB process will be triggered. Here “unconfigured” has the same meaning as for ZTP: no changes made to the ogconfig data store.

Note: Setting the root password on first log in counts as a config change.

The following manifest.og keys are implemented. This provides image installation, Lighthouse enrollment, and arbitrary script execution:

manifest.og contains <key>=<value> pairs. Recognized keys are:

image : Firmware image file name on the USB device's filesystem that will be flashed after boot once the image is validated

script : Configuration script to run

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

23.10.3	Custom Control Codes for Serial Ports	82
---------	---------------------------------------	----



external_endpoints : List of additional "address:port" endpoints to fall back to when enrolling

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

EULA AND GPL

The current Opengear End-User License Agreement and the GPL can be found at <http://opengear.com/eula>.

23.10.3	Custom Control Codes for Serial Ports	83
---------	---------------------------------------	----

ACCESS MENU

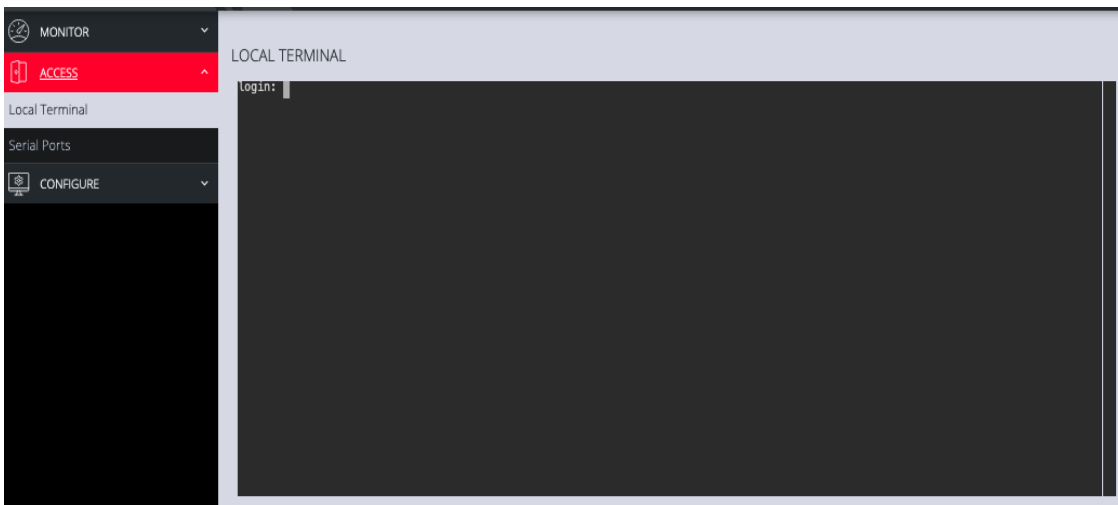
The ACCESS menu lets you access the Console Manager via a built-in Web Terminal. It also provides SSH and Web Terminal access to specific ports.

LOCAL TERMINAL

ACCESS > Local Terminal

The Console Manager includes a web-based terminal. To access this bash shell instance:

1. Select **ACCESS > Local Terminal**



2. At the login prompt, enter a username and password.
3. A bash shell prompt appears.

This shell supports most standard bash commands and also supports copy-and-paste to and from the terminal.

To close a terminal session, close the tab, or type exit in the Web Terminal window. The session will timeout after 60 seconds.

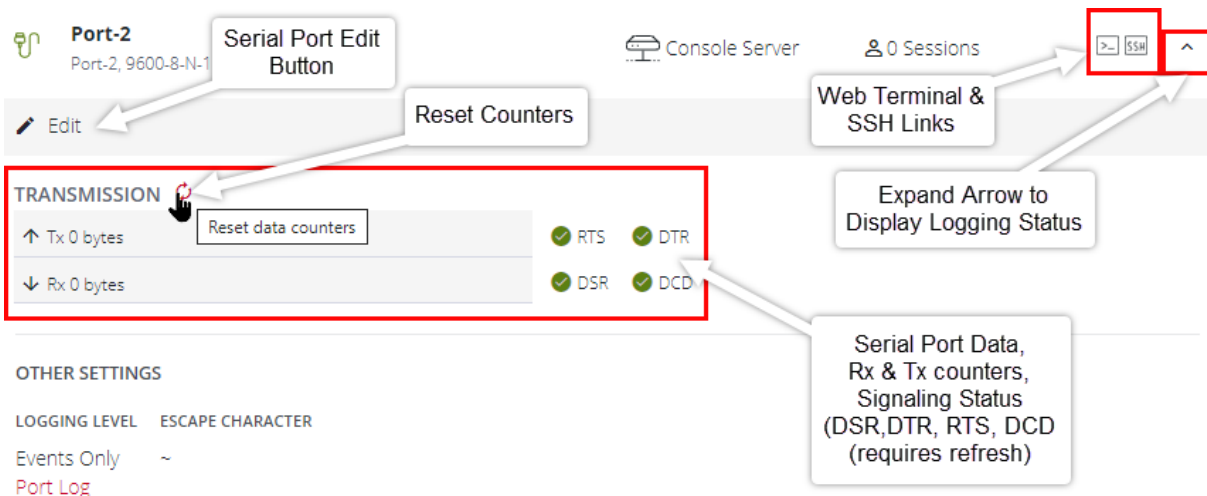
Tip: The default for the CLI session timeout is “never” (value of 0), however, the Web session timeout defaults to 20min. The web session time-out will kill the CLI session even though the CLI session itself is set to “never”.

ACCESS SERIAL PORTS

ACCESS > Serial Ports

Tip: Ensure you are on the **ACCESS > Serial Ports** page and not the similar **CONFIGURE > Serial Ports** page.

The **ACCESS > Serial Ports** page allows you to quickly locate and access specific ports via Web Terminal or SSH link shown in the image below.



Click the **Expand arrow** to the right of the port to see the Port Logging status or access the port **Edit** button, which is a link to the **CONFIGURE > Serial Ports** page. (ogcli: `ogcli get ports/ports_status`).

23.10.3	ACCESS Menu	85
---------	-------------	----

The following information is displayed under **Access > Serial Ports** when the individual serial ports are expanded:

- Rx byte counter (counter reset requires 'Admin' or 'port config' rights)
- Tx byte counter (counter reset requires 'Admin' or 'port config' rights)
- Signaling information (DSR, DTR, CTS (see tip), RTS and DCD)

Tip: CTS information is not displayed in the UI but is available via the ogcli query `ogcli get ports/ports_status`.

- Logging information.

QUICK SEARCH

To find a specific port by its port label, use the **Quick Search** form at the top-right of the **ACCESS > Serial Ports** page.

Ports have default numbered labels. You can edit the port label for a given serial port under **CONFIGURE > Serial Ports**. Click the **Edit** button to open the **EDIT SERIAL PORT** page.

ACCESS USING WEB TERMINAL OR SSH

To access the console port via the Web Terminal or SSH:

1. Locate the particular port on the **ACCESS > Serial Ports** page and click the expand arrow.
2. Click the **Web Terminal** or SSH link for the particular port.

- Choosing **Web Terminal** opens a new browser tab with the terminal.
- Choosing **SSH** opens an application you have previously associated with SSH connections from your browser.

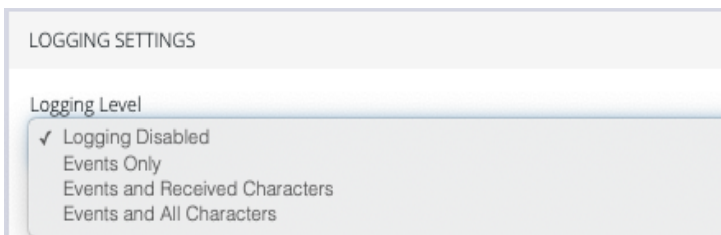
Note:MS Windows does not connect to puTTY by default. You may need to install the WinSCP program to launch puTTY from the Opengear WebGUI SSH Serial Port button.

SERIAL PORT LOGGING

The port logging facility and severity associated with the serial port logs is controlled and set at the **Configure > Services > Syslog > Global Serial Port Settings** page.


There is a separate setting to enable sending of serial port logs to remote side.


Note:Serial port logging is disabled by default. The logging level for each serial port is set at Logging Settings in **Configure > Serial Ports > Edit** .



DISPLAY PORT LOGS

Tip: The log is accessed by clicking the **Port Log** link on the **ACCESS > Serial Ports** page. The link is only available when port logging is enabled.

 **Port-1**
Port-1, 9600-8-N-1-X2

 Edit

LOGGING LEVEL	ESCAPE CHARACTER
Events Only	~
Port Log	

Port Log Link →

CONFIGURE MENU

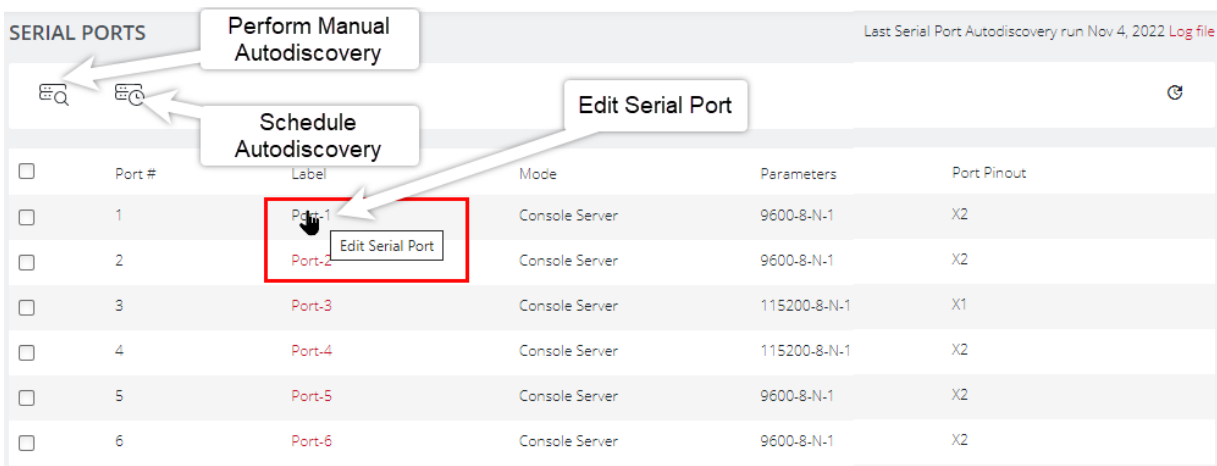
This section provides step-by-step instructions for the menu items under the CONFIGURE menu.

CONFIGURE SERIAL PORTS

CONFIGURE > Serial Ports

Tip: Ensure you are on the **CONFIGURE > Serial Ports** page and not the similar **ACCESS > Serial Ports** page.

Navigate to **CONFIGURE > Serial Ports**; a list of serial ports is displayed. On this page you can configure and edit specific ports. Click the **Edit** button (pencil icon) to the right of the port to display the port editing page.



<input type="checkbox"/>	Port #	Label	Mode	Parameters	Port Pinout
<input type="checkbox"/>	1	Port-1	Console Server	9600-8-N-1	X2
<input type="checkbox"/>	2	Port-2	Console Server	9600-8-N-1	X2
<input type="checkbox"/>	3	Port-3	Console Server	115200-8-N-1	X1
<input type="checkbox"/>	4	Port-4	Console Server	115200-8-N-1	X2
<input type="checkbox"/>	5	Port-5	Console Server	9600-8-N-1	X2
<input type="checkbox"/>	6	Port-6	Console Server	9600-8-N-1	X2

EDIT SERIAL PORTS

From the **Configure > Serial Ports** page, click the **Port label** text in the Label column. The **Edit Serial Port** page is displayed.

Edit Serial Port Properties		
Field	Options	Definition
Label	Default or Custom	The serial port unique identifier. This can be used to locate this port using the Quick Search form on the ACCESS > Serial Ports page.
Mode	Disabled Console Server Local Console	Console Server mode allows access to a downstream device via its serial port. Local Console mode allows access to the OM device's console through a serial port.
Port Pinout	Fixed - X2 Cisco Straight	The pin-out type is fixed on the CM8100.
Port Pinout CM8100-10G	Selectable - X2 Cisco Straight	The pin-out type is software selectable on the CM8100-10G
Baud Rate	Baud rate	Select the Baud rate expected for this port. From 50 to 230,400 bps.

Data Bits	Integer	The data bit length for character.
Parity	None, Odd, Even, Mark, Space.	The parity type for character.
Stop Bits	1, 1.5, 2	The Stop bit length used in character.
Escape Character	~	The character used for sending OOB Shell commands.
LOGGING SETTINGS		
Logging Level	Disabled Events Only Events & Received Characters Events & All Characters	Specify the level of detail you require in the logs. Logs may also be sent to a Syslog server. Other settings to consider are: "GLOBAL SERIAL PORT SETTINGS" under Services > Syslog. "Send Serial Port Logs" under Services > Syslog > Add Syslog Server
PORT IP ALIASES		
IP Address	Alias IP Address and interface type.	Allocate an IP address for dedicated access to a specific serial port.

ASSIGNING UNIQUE IP ADDRESSES FOR EACH CONSOLE PORT

Note:For further information about assigning unique IP addresses for each console port see the Zendesk article [Assigning Unique IP Addresses For Each Console Port](#) .

23.10.3	CONFIGURE Menu	92
---------	----------------	----

CONFIGURE SINGLE SESSIONS FOR PORTS

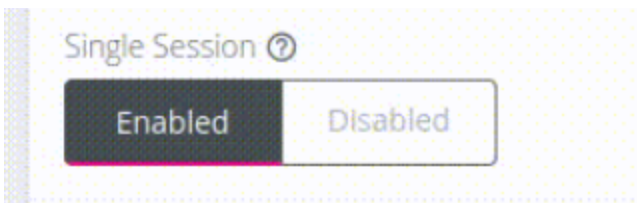
Single Session Port Config, or *Single Session* is a feature that can be enabled on a given port to prevent multiple users from connecting to that port or limit the port to a single concurrent connection. This feature is port-specific and is disabled by default. This feature needs to be enabled on a port-by-port basis. It can be enabled on all types of serial ports (including USB).

Similarly to config shell, a single session must be enabled/disabled on a port by port basis, currently it cannot be enabled on all ports.

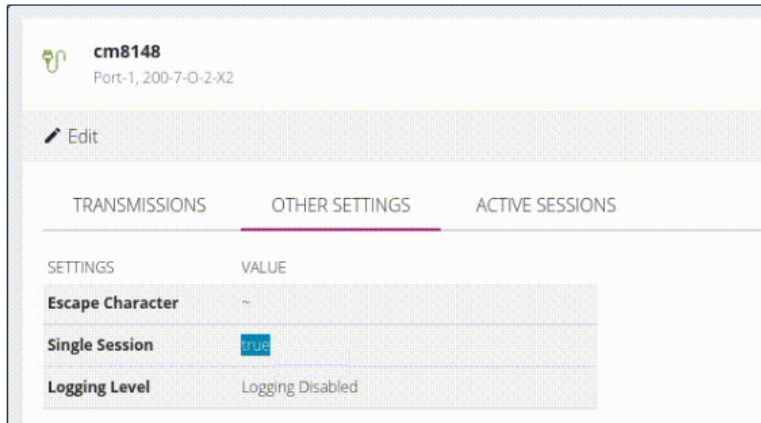
SINGLE SESSION ENABLED

IN THE WEB UI

Single Session can be viewed and configured in the Web UI. It is enabled (or disabled) in the configure page for a given serial port. The buttons to connect to a serial port are automatically disabled when the feature is enabled and the session is in use.



You can also confirm the session in the `access/serial_ports` page, and also see if the feature is enabled.

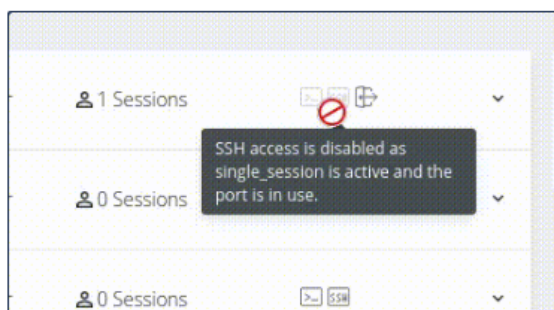


When the Single Session feature is enabled and the port is in use, if a subsequent user attempts to connect to the port, the connection is declined and the second user will receive the message:

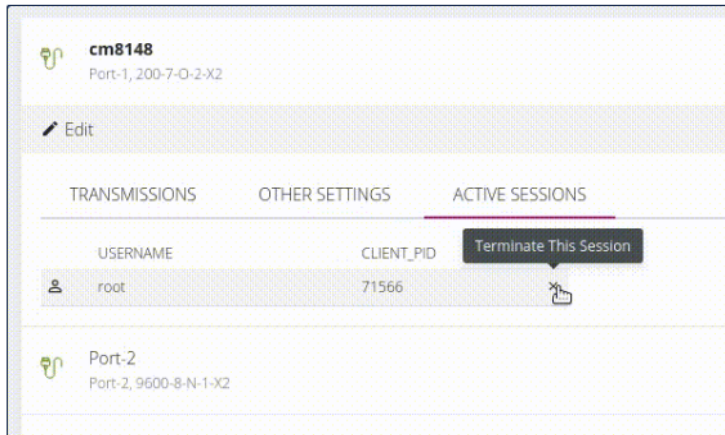
```
Unable to connect. Another session is currently active.  
Please disconnect from the current session before attempting to  
connect again.
```

The pmsHELL will exit, and the user who tried to connect will not have seen the first user's session. Everything they have done will remain confidential.

The single session is indicated next to the user



If necessary, a user's single session can be terminated with the **Terminate all sessions** button which is shown beside individual users. This will re-enable the **Single Session** button and allow you to connect.



IN CONFIG SHELL

The Single Session feature can be enabled or disabled by editing the `single_session` field in a given port. When a user port level administration access is logged in via `pmshell`, the port configuration menu can be accessed via any port by pressing the escape character (`~` by default) followed by `c (~c)`.

You can access a port with the following commands, the following example will access "Port 1":

```
config: port
config(port): port01
```

The port configuration might look like this below. You can see for this port, `single_session` is set to `false`, so the feature is disabled:

```
config(port port01): show
Entity port item port01
  baudrate 9600
  databits 8
  escape_char ~
  label Port-1
  logging_level disabled
```

```
mode consoleServer
parity none
pinout X2
portnum 1
single_session false
stopbits 1
control_code (object)
  break ""
  chooser ""
  pmhelp ""
  portlog ""
  power ""
  quit ""
ip_alias (array)
```

The feature is enabled by typing `single_session true`, then apply the change.

```
config(port port01): single_session true
config(port port01): apply
Updating entity port item port01.
config(port port01): show
Entity port item port01
  baudrate 9600
...
single_session true
...
ip_alias (array)
```

SINGLE SESSION BEHAVIOR

The following table describes single session feature behavior in various circumstances.

Q.	What occurs if users are connected to the port with the feature disabled, then the feature is enabled while users are still connected?
A.	Users who are already connected will continue to be able to use the port. If they leave, they will not be able to rejoin (unless there are no active sessions). Their current session will continue as normal, however, their session can be manually terminated from config shell (config(port_session):) or from the Web UI from the Access/Serial Ports page.
Q.	What if a user needs to be removed from a port?
A.	Administrators can remove the right for a given user to access a port. They can also manually remove them from the port in the config shell (config(port_session):) or the Web UI from the Access/Serial Ports page.
Q.	What if someone tries to join a port that is already in use?
A.	The user who tries to join will be prevented from doing so and receive a notification. The person currently using the port will be unaffected and not be aware of the attempt.
Q.	Is there a way to enable the feature for every port?
A.	Currently, the feature must be enabled/disabled on a port-by-port basis.
Q.	What if I enable this port on localConsole mode?
A.	The feature is ignored on local console mode and is only active for consoleServer mode. It also remains ignored if the port mode is set to

disabled.

23.10.3	CONFIGURE Menu	98
---------	----------------	----

AUTODISCOVERY

The Autodiscovery feature attempts to discover the host name of connected devices, this uses the hostname of the device used to set the port label, so as to set it as the port label of each serial port. This can save the need to manually provide hostnames during setup.

Autodiscovery will attempt to discover port settings even if the hostname discovery fails. The first discovery run uses currently configured port settings such as the current baud rate, etc. Thereafter, it will fetch or use a single set of pre-configured credentials to login and discover the hostname from e.g. the OS prompt, for devices that do not display hostname pre-authentication.

Syslogging enhancement assists in the diagnosis of common issues (for example, no comms or, hostname failed validation). Autodiscovery does not collect a hostname when there is a communication issue between the console server and the target device. The logs are saved for the last-run instance of autodiscovery.

The UI displays error messages and logs with the reason for auto-discovery failure, for example:

- Authentication failed.
- Communication issue with the target device.
- Password to renew before being able to authenticate to the target device.
- Abnormal characters or strings detected.

Autodiscovery has been enhanced to discover baud rate and pinout. The UI now indicates if ports are scheduled for discovery.

The **Serial Ports** page also allows you perform an Autodiscovery on selected ports. Autodiscovery of console ports attempts to set the port label by setting the baud rate to various rates (in the following order): 9600, 115200, 38400, 19200, and 57600.

Tip: Autodiscovery on other Baud rates may be done by manually running the `port_discovery` script from the Web Terminal.

Autodiscovery may be done manually by clicking **Perform Autodiscovery**.

AUTODISCOVERY ENHANCEMENTS

From the 22.11 release, the following parameter enhancements have been added to the `port_discovery` script which can be configured via the REST API or CLI:

- `--username` and `--password`
- `--apply-config` and `--no-apply-config`
- `--auth-timeout`
- `--hostname-pattern`

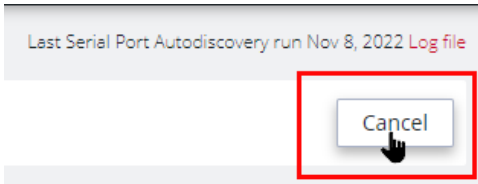
The `--username` and `--password` options can also be configured via the UI under *Optional Credentials*.

If the values are provided (optional), they will be used to attempt login to obtain the hostname to a downstream serial device. You can only specify a single username and/or password to try on all devices.

Optional Credentials ⓘ

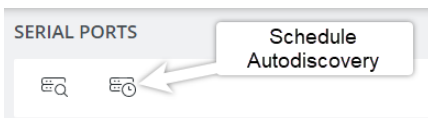
CANCEL AUTODISCOVERY

Port Autodiscovery may be canceled *while running* by clicking on the **Cancel** button at the top-right of the Serial Ports window of the UI.



SCHEDULE AUTODISCOVERY


Autodiscovery can be scheduled periodically as required by clicking the **Schedule Autodiscovery** button in the **Serial Ports** window.



The **Schedule Autodiscovery** window allows you to select the ports and specify a time and period for port detection to run. Activate the schedule by clicking on the **Enabled** button.


The Serial Port Autodiscovery Page:


SCHEDULE SERIAL PORT AUTODISCOVERY

Status 


Enabled Disabled

CONFIGURE SCHEDULE

Repeat at 

Advanced Configuration 

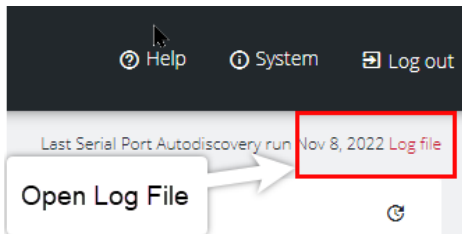
SELECT PORTS

 Serial Port Autodiscovery will be only performed on ports in Console Server Mode.


<input type="checkbox"/>	Port #	Label	Mode	Parameters	Port Pinout
<input checked="" type="checkbox"/>	1	Port-1	Console Server	9600-8-N-1	X2
<input checked="" type="checkbox"/>	2	Port-2	Console Server	9600-8-N-1	X2
<input checked="" type="checkbox"/>	3	Port-3	Console Server	115200-8-N-1	X1
<input checked="" type="checkbox"/>	4	Port-4	Console Server	115200-8-N-1	X2

RETRIEVE PORT DISCOVERY LOGS

At the top-right of the UI window, click on the **Log File** red text to retrieve the port discovery logs or by clicking on the **View Logs** red text in the **autodiscovery running** banner.



SERIAL PORTS

 **Serial Port Autodiscovery is running**
The task times can vary based on latency, hardware, and number of ports. **View Logs**

Port Discovery Log File Example:

SERIAL PORT AUTODISCOVERY LOGS - LAST COMPLETED RUN

```
[main] Starting discovery with 9600 baud and X2 pinout on preconfigured port 4
[port4] 2022-11-08T07:47:16+0000 Discovery starting
[port4] Checking port readiness
[port4] No device discovered
[main] Starting discovery with 9600 baud and X2 pinout
[main] Skipping duplicate test: port 4, baud 9600, pinout X2
[main] Starting discovery with 115200 baud and X2 pinout
[port4] 2022-11-08T07:48:09+0000 Discovery starting
[port4] Checking port readiness
[port4] No device discovered
[main] Starting discovery with 38400 baud and X2 pinout
[port4] 2022-11-08T07:49:00+0000 Discovery starting
[port4] Checking port readiness
[port4] No device discovered
[main] Starting discovery with 19200 baud and X2 pinout
[port4] 2022-11-08T07:49:51+0000 Discovery starting
[port4] Checking port readiness
[port4] No device discovered
```

DISPLAY OPTIONS

Number of Log Lines ⓘ

Apply

LOCAL MANAGEMENT CONSOLES

CONFIGURE > Local Management Consoles

This feature allows administrators to log in and configure the CM via the RJ-45 ports on the device. Not accessible by USB.

To edit the settings of a local management console:

1. Navigate to **CONFIGURE > Local Management Consoles**. Here you'll see a list of consoles.
2. Locate the console you want to manage, then, click on the **Edit Management Console Port** button (pencil icon) under **Actions**.
3. On the **Edit Local Management Console** page you can set the parameters for:
 - **Baud Rate**
 - **Data Bits**
 - **Parity**

23.10.3	CONFIGURE Menu	103
---------	----------------	-----



- **Stop Bits**
- **Terminal Emulation**
- Enable or disable **Kernel Debug Messages**
- Enable or disable the selected **Management Console**

Note:Enabling **Kernel Debug Messages** can only be applied to a single serial management console.

To disable a local management console:

1. Click **CONFIGURE > Local Management Consoles**.
2. Click on the **Disable Management Console Port** button under **Actions** next to the console you wish to disable.

LIGHTHOUSE ENROLLMENT

CONFIGURE > Lighthouse Enrollment

Opengear appliances can be enrolled into a Lighthouse instance, providing centralized access to console ports, automation, and central configuration of Opengear devices.

Lighthouse central management uses a persistent, public key authenticated SSH tunnels to maintain connectivity to managed console servers.

All network communications between Lighthouse and each console server (e.g. access to the web UI), and the console server's managed devices (e.g. the serial consoles of network equipment), is tunneled through this SSH management tunnel.



The below Zendesk articles and user guide contain further information about Lighthouse Enrollment:

[Manual enrollment using UI or CLI](#)

[How do I add Nodes to Lighthouse](#)

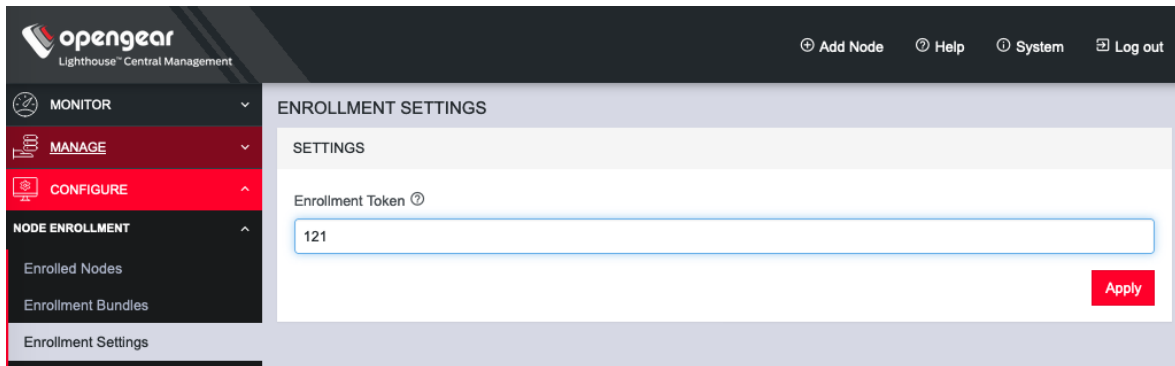
[Lighthouse User Guide](#)

23.10.3	CONFIGURE Menu	105
---------	----------------	-----

MANUAL ENROLLMENT USING UI

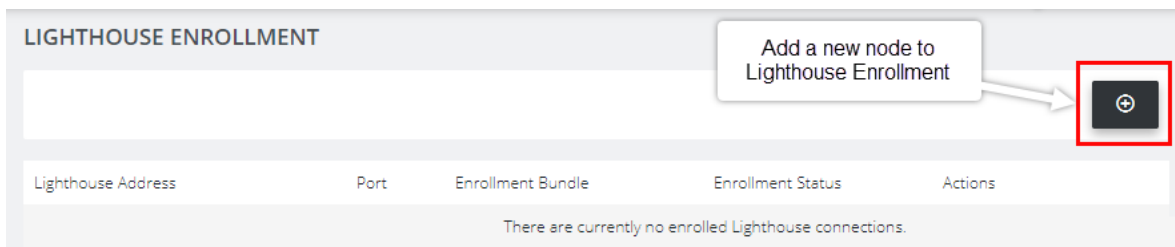
Note: To enroll your Console Manager to a Lighthouse instance, you must have Lighthouse installed and have an enrollment token set in Lighthouse.

1. In Lighthouse. Set a CM enrollment token, click on **CONFIGURE > NODE ENROLLMENT > Enrollment Settings** page, and enter an **Enrollment Token**.

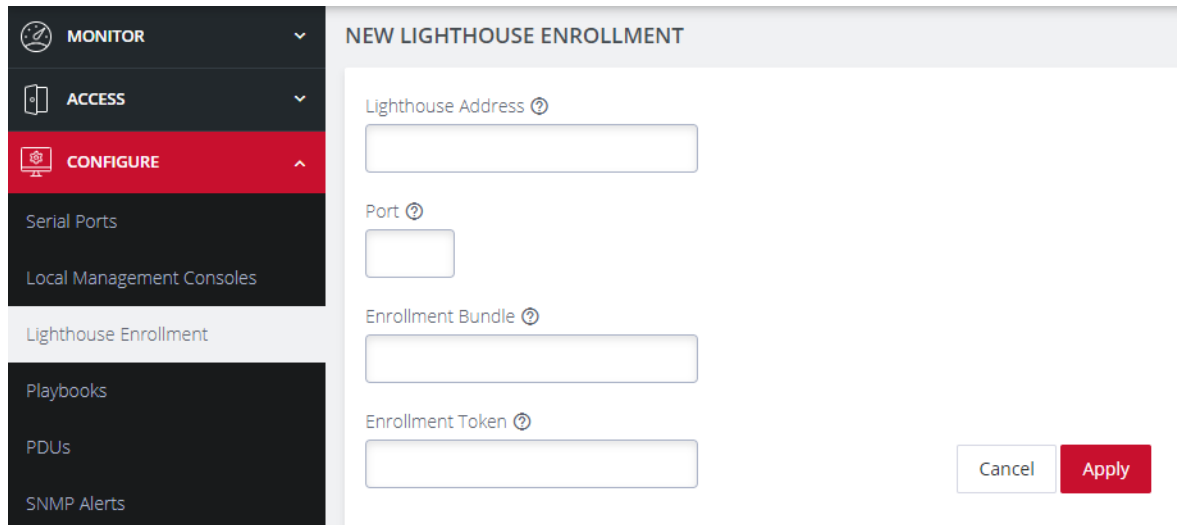


Tip: The same token will be entered in the NEW LIGHTHOUSE ENROLLMENT page of the Console Manager.

2. Enroll your Console Manager in this Lighthouse instance:
Click **CONFIGURE > Lighthouse Enrollment**
3. Click on the **Add Lighthouse Enrollment** button on the top-right of the page.
The **New Lighthouse Enrollment** page opens.



4. Enter the IP address or fully qualified domain name of the Lighthouse instance and the **Enrollment Token** you created in Lighthouse. Optionally enter a **Port** and an **Enrollment Bundle** (see the [Lighthouse User Guide](#) for more information about Bundling).



The screenshot shows the 'NEW LIGHTHOUSE ENROLLMENT' form in the OpenGear configuration interface. The left sidebar is expanded to the 'CONFIGURE' section, with 'Lighthouse Enrollment' selected. The main form area contains four input fields: 'Lighthouse Address', 'Port', 'Enrollment Bundle', and 'Enrollment Token'. Each field has a help icon (a question mark in a circle). At the bottom right of the form are two buttons: 'Cancel' and 'Apply'.

5. Click the **Apply** button. A flag will confirm the enrollment.

Note: Enrollment can also be done directly via Lighthouse using the Add Node function. See the Lighthouse User Guide for more instructions on enrolling OpenGear devices into Lighthouse.

MANUAL ENROLLMENT USING THE CLI

For complete instructions on Lighthouse Enrollment via the CLI please refer to this link: [Manual enrollment using UI or CLI](#) .

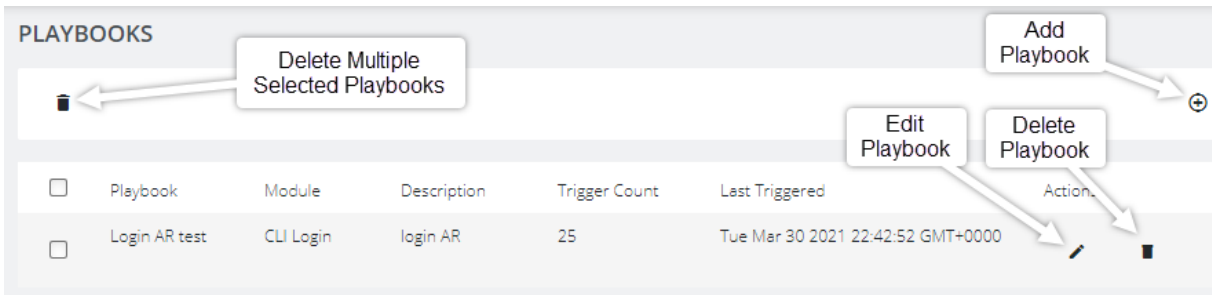
PLAYBOOKS

[CONFIGURE > Playbooks](#)

23.10.3	CONFIGURE Menu	107
---------	----------------	-----

Playbooks are configurable systems that periodically check if a user-defined **Trigger** condition has been met. Playbooks can be configured to perform one or more specified **Reactions** when a specific trigger event occurs.

The Playbook Landing Page:



CREATE OR EDIT A PLAYBOOK

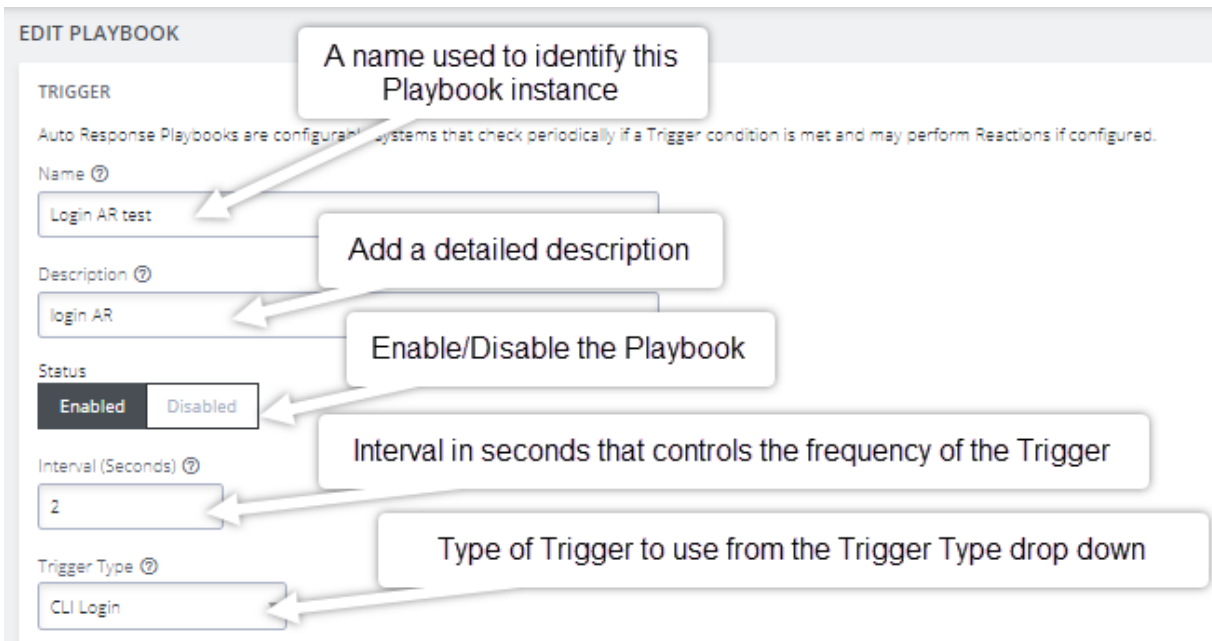
[CONFIGURE > Playbooks > Add Playbook](#)

To create a new Playbook:

Navigate to the **Configure > Playbooks** page.

Click the **Add Playbook** button (top-right) to create a new **Playbook**. The **Edit Playbook** page is displayed. Complete the required Playbook setup information as detailed in the following procedures.

TRIGGER SECTION:



EDIT PLAYBOOK

TRIGGER

Auto Response Playbooks are configurable systems that check periodically if a Trigger condition is met and may perform Reactions if configured.

Name ⓘ
Login AR test

Description ⓘ
login AR

Status
 Enabled Disabled

Interval (Seconds) ⓘ
2

Trigger Type ⓘ
CLI Login

Callouts:
- A name used to identify this Playbook instance (points to Name)
- Add a detailed description (points to Description)
- Enable/Disable the Playbook (points to Status)
- Interval in seconds that controls the frequency of the Trigger (points to Interval)
- Type of Trigger to use from the Trigger Type drop down (points to Trigger Type)

1. Enter a unique **Name** for the **Playbook** that reflects its purpose.
2. Add a detailed **Description** that will help others to understand what it does.
3. Select **Enabled** to activate the **Playbook** after you have created it.
4. Enter an **Interval** in seconds to control the frequency that the **Trigger** will be checked.
5. Choose the type of **Trigger** to use from the **Trigger Type** drop down.

Tip: See the Trigger Type table on the following page for additional trigger type information.

TRIGGER TYPES:

Trigger	Reaction Description
CLI Login	Triggers upon Login or Logout events. Select either or both.
CLI Login Failure	Monitor the terminal and trigger on failed user login attempts.
Cell Connection	Triggered whenever the cellular connection state changes. This Trigger type is only compatible with cellular units.
Cell Message	Triggered when an SMS message that matches the user-defined message pattern. Cellular units only.
Cell Signal Strength	Triggered if the cellular signal strength moves below a user-defined percentage.
Curl	Periodically attempts to perform a HTTP request using curl and triggers the Playbook reaction based on the results.
Custom Command	Periodically runs a custom Shell command and triggers the Playbook reaction upon failure.
Load	Monitors the system load average and triggers the Playbook if it breaches the user-defined acceptable range.
Memory Usage	Triggered if the system memory usage exceeds the user-defined percentage threshold.

Network Settings	Monitors network interfaces for specific attributes and triggers a user-defined response when they change.
Ping	Periodically pings an address and triggers a user-defined response upon failure.

Continued...

Trigger	Description
Serial Login	Monitors selected serial ports and triggers a user-defined reaction upon user login and logout events.
Serial Pattern	Monitors serial ports and triggers a reaction when data matching a pattern is received on specific ports.
Serial Signal	Monitors selected serial ports and triggers when signals are changed.

REACTION SECTION:

In this section you customize the response to the Trigger that you created.

1. Clicking on each **Reaction** opens a custom screen to provide necessary information.

REACTION
Reactions are configurable actions taken when a Trigger condition is met.

Select the required response to a trigger

Send SMS Custom Command **Send Text** Slack SNMP

Name ⓘ
execute script

The name used to identify this Reaction instance

Shell Command ⓘ
/bin/r-login.sh

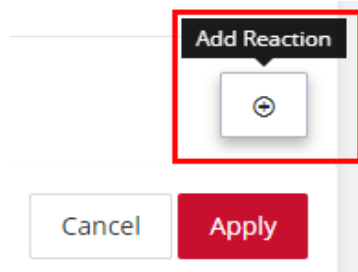
Insert a shell command that will run when an event type is triggered. (Command types will vary depending which Reaction is selected).

Timeout ⓘ
10

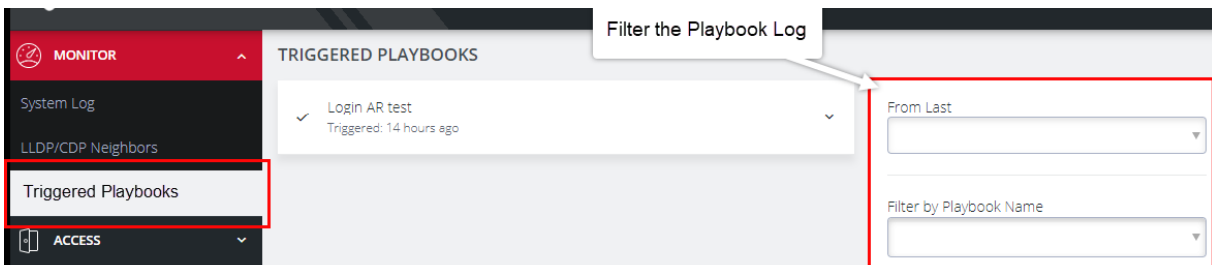
The time, in seconds, to wait for a response.

Continued...

2. To create additional Reactions, click the **Add Reaction** button.



3. When you are finished, click **Apply**. A banner confirms that the Playbook settings are saved, if the Playbook is **Enabled** it is activated.
4. To monitor current **Playbooks**, click on the **Monitor > Triggered Playbooks** menu (shown below). Select the time period if desired and filter by **Name** of **Playlist** to view any that have been triggered.

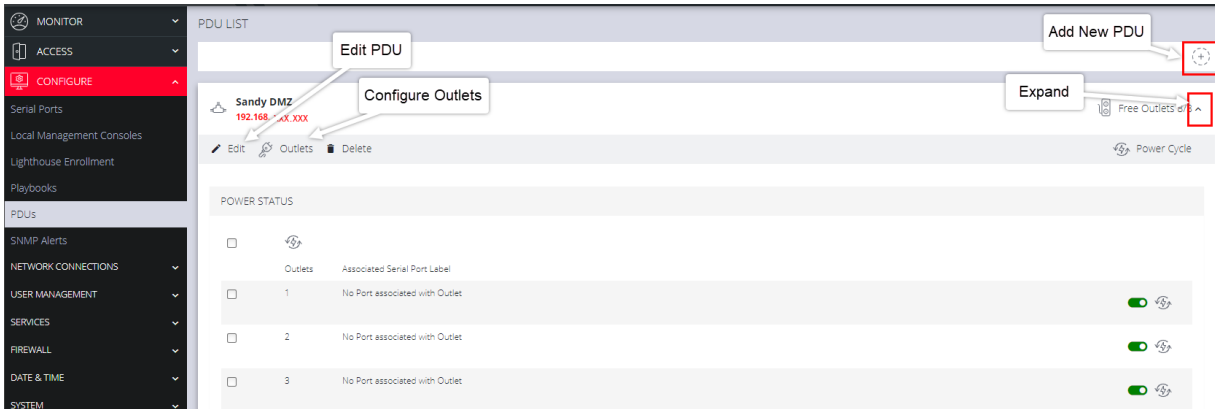


PDUS

CONFIGURE > PDUs

One or more Power Distribution Units (**PDUs**), both **Local** and **Remote** can be monitored. To add information for a **PDU**, select **Configure > PDUs**.

23.10.3	CONFIGURE Menu	113
---------	----------------	-----



ADD AND CONFIGURE A PDU

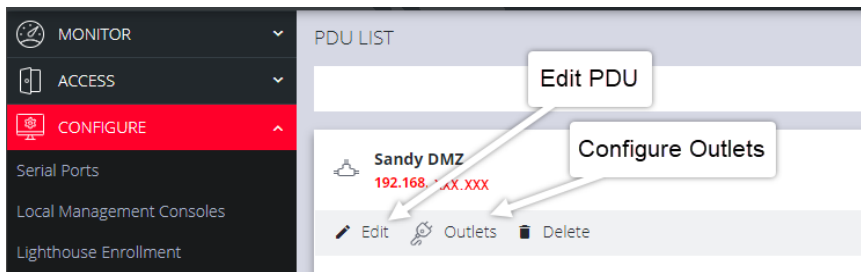
PDU configuration definitions are provided in the on the "[PDU Settings Table](#)" on the [next page](#).

1. In the PDU List page, click the **Add New PDU** button. The **Edit** page opens.
2. Enter a meaningful **Label** that will easily identify this **PDU**.
3. Select the **Monitor** checkbox.
4. Select **Local** or **Remote**.

Note:Note that **Local** or **Remote** have different settings forms.

5. Complete the **Local** or **Remote** settings in accordance with the "[PDU Settings Table](#)" on the [next page](#).

- Click on the **Configure Outlets** link, assign a port for each of the PDUs' ports and enter a meaningful name for each outlet.



- When you are finished, click **Apply**. A green banner confirms your settings.

PDU SETTINGS TABLE

PDU Settings	
Label	Enter a meaningful label that will easily identify the individual PDU .
Monitor	Click to check this box to monitor the outlet's status.
Mode	Note that (Local or Remote have different settings forms).
Driver	Select the appropriate driver compatible with this PDU.
Local Mode Only	
Port	The serial port that the PDU is connected to.
Username	Enter the Username to use when connecting.

Password	User password to use when connecting to the device.
----------	---

The table is continued on the following page...

23.10.3	CONFIGURE Menu	116
---------	----------------	-----

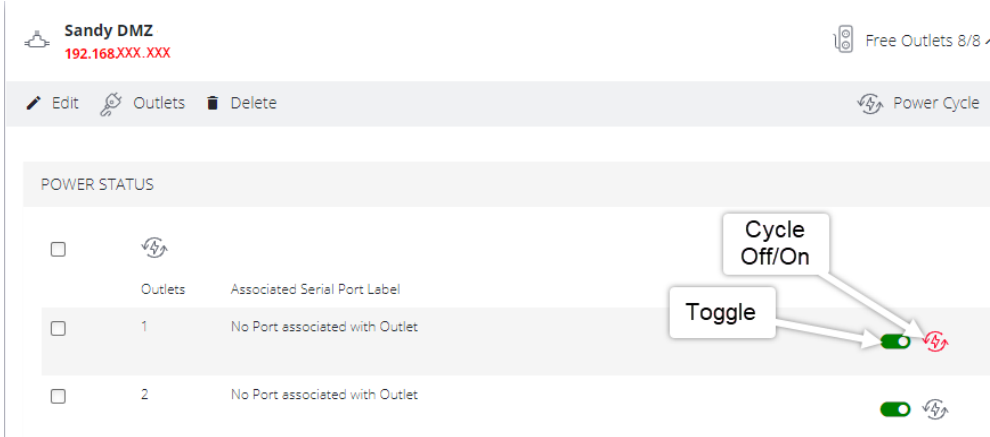
Continued...

Remote Mode Only	
Address	The remote address of the PDU.
SNMP Protocol	Click the drop-down arrow and select the correct transport protocol used to communicate with the PDU. The default value is UDP.
Version	The version of SNMP to use, V1, V2c and V3 are supported. The default value is V1.
Community	Enter a group name authorized to communicate with the device for SNMP versions 1 and 2c.

After you have created **PDU**s, you can **Edit** or **Delete** them from the **Configure > PDU**s page.

PDU OPERATION

After the PDU has been created and configured, PDU operation is simple. For any PDU that has Monitoring set to **Enabled**, the **Toggle** on/off switch will power-on or power-off the PDU, and the **Cycle** button cycles the PDU through a power-down and power-up cycle.



SYSTEM ALERTS

[CONFIGURE](#) > [System Alerts](#) > [General/Power/Temperature/Networking](#)

Tip: For more detailed information about configuring SNMP Alerts see the individual topic pages that follow.

System Alert Managers can be added or deleted under [Configure](#) > [Services](#) > "[SNMP Alert Managers](#)" on [page 287](#), for the following:

- **General:** Covers notification for the following causes.
 - **Authentication:** Notifies when a user attempts to log in via SSH, REST API, Web GUI, or the device's serial ports. An alert is sent regardless of whether the log in has succeeded or failed.
 - **Configuration Change:** For changes that occur to the system configuration.
- **Power:** When voltage SNMP alerts are enabled, network operators are immediately notified should the PSU begin operating outside design tolerances.

- **Temperature:** When system temperature alerts are enabled, network operators are immediately notified should the system begin operating outside user-defined tolerances.
- **Networking (Cell Signal Strength):** Be notified when cell signal strength leaves or re-enters the selected range, or when the network link state changes. A slider adjusts the upper and lower signal strength.

Tip: Manage the system settings on the **CONFIGURE > System Alerts > System Alerts** pages.

SYSTEM ALERTS - GENERAL

AUTHENTICATION

[CONFIGURE > System Alerts > General > Authentication](#)

Notifies when a user attempts to log in via SSH, REST API, or the device's serial ports. An alert is sent regardless of whether the log in has succeeded or failed.

1. Navigate to **Configure > System Alerts > General > Authentication**.
2. Click on the **Enabled** button to activate the function.
3. Click **Apply**. The **Details Saved** banner confirms your settings.

CONFIGURATION

[CONFIGURE > SNMP Alerts > System > Configuration Change](#)

Notifies of changes that occur to the system configuration.



1. Navigate to Configure > SNMP Alerts > System > Configuration.
2. Click on the **Alerting** button to activate the function.
3. Click **Apply**. The **Details Saved** banner confirms your settings.

SYSTEM ALERTS - POWER

[Configure > System Alerts > Power > Voltage](#)

The PSU is one of the most critical part of the Console Manager so it is essential to ensure that the PSU is operating within its design tolerances.

When voltage SNMP alerts are enabled, network operators are immediately notified of PSU failures (subject to network connectivity and latency). Should the PSU begin operating outside design tolerances, PSU-related SNMP Alerts will trigger an alert for the following conditions:

- Output DC voltage of both PSUs

If the voltage drops too low, it risks the Console Manager going into brown-out state. If it gets too high, it can damage components.

System generated SNMP Alerts send SNMP traps to a remote SNMP manager which alerts the user of system events. The Console Manager can send network, power and system events to the remote SNMP manager.

Tip: The Console Manager can send network, power and system events to the remote SNMP manager.

ENABLE POWER SUPPLY SYSLOG ALERTS

[Configure > System Alerts > Power > Voltage](#)

23.10.3	CONFIGURE Menu	120
---------	----------------	-----



The System Voltage Range alert sends an alert when the system reboots or the voltage on either power supply leaves or re-enters the fixed voltage range between 11.4V to 12.6V (SNMP) (or 11V to 13V Syslog).

1. Navigate to **Configure > System Alerts > Power > Voltage**.
2. Click on the **Enabled** button to activate the function.

Note: The **Disabled** button de-activates the power syslog function and power alerts will be stopped until activated again

Syslog Alert Severity

Configure > Syslog > Add Syslog Server

3. For **Power Lost** alert, click the drop-down list and select the severity level required (default level is **3 - ERROR**) when power level is outside the pre-set range.
4. For **Power Restored** alert, click the drop-down list and select the severity level required (default is **6 - INFO**) after an error condition has been fixed.
5. Click **Apply**. The **Details Saved** banner confirms your settings.

When an event occurs that causes the voltage range on any power supply to leave or re-enter the configured voltage range, it will cause an SNMP alert to be triggered. The alert will report the event type and identity and status of the PSU, as in the example below.

```
Nov 03 06:09:35 om2232 system-alerts[850]: Redundant Supply Active (PSU0 online, PSU1 online)
```

```
Nov 03 07:05:02 om2232 system-alerts[850]: Redundant Supply Inactive (PSU0 offline, PSU1 online)
```

```
Nov 03 07:05:05 om2232 system-alerts[850]: Redundant Supply Active (PSU0 online, PSU1 online)
```

To view log severity messages locally, use the journal tool command `journalctl -f -u alert-logger -o verbose` where: f = follow. Check the alert-logger using the `systemctl status alert-logger` command.

SYSTEM ALERTS - NETWORKING (CONNECTION STATUS)

[Configure > SNMP Alerts > Networking > Network Connection Status](#)

The alert related to this functionality is the Network Connection Status which sends an alert when cell signal strength leaves or re-enters a user-defined range, or, when the network link state changes. A slider adjusts the upper and lower signal strength limits.

CONFIGURE SIGNAL STRENGTH ALERTS

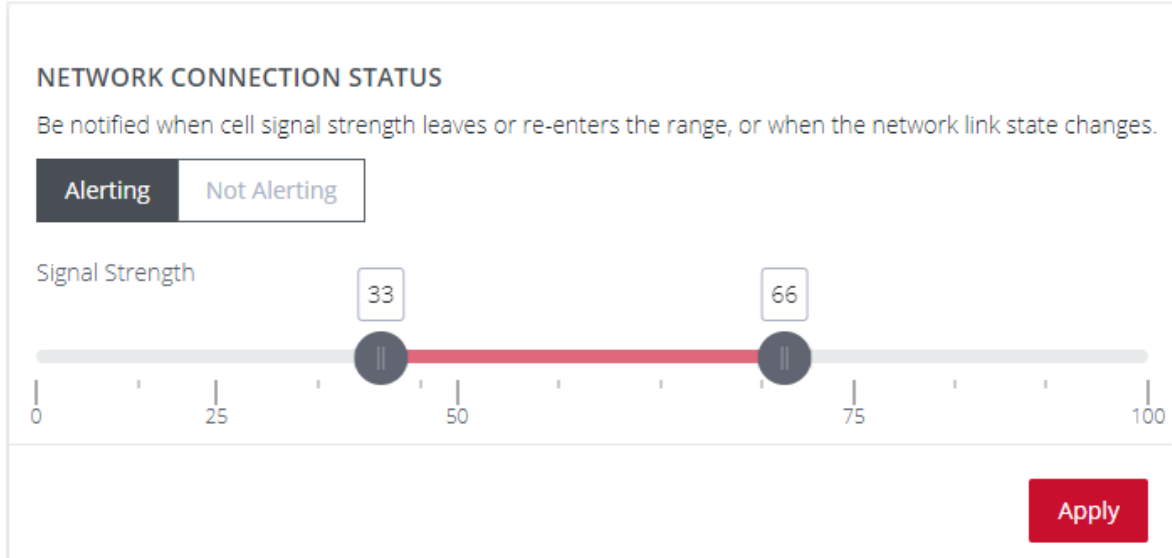
[Configure > SNMP Alerts > Networking > Network Connection Status](#)

To set the Network Connection Status signal strength boundaries:

1. Navigate to [Configure > SNMP Alerts > Network Connection Status > Signal Strength](#) page.
2. Click on the **Alerting** button to activate the function, this also activates the user-defined range sliders.
3. Click+Drag the signal strength range limiters to the required upper and lower limits.

Note: The **Not Alerting** button de-activates the function and signal strength alerts will be stopped until activated again.

4. Click **Apply**. The **Details Saved** banner confirms your settings.



NETWORK CONNECTION STATUS

Be notified when cell signal strength leaves or re-enters the range, or when the network link state changes.

Alerting Not Alerting

Signal Strength

0 25 33 50 66 75 100

Apply

When an event occurs that causes the signal strength to re-enter the user-defined range, an SNMP alert will be triggered.

In the above image, if any anomaly occurs that causes the signal strength to drop below 33 or above 66, an SNMP alert will be triggered.

NETWORK CONNECTIONS

[CONFIGURE > NETWORK CONNECTIONS](#)

The **Network Connections** menu contains the **Network Interfaces**, **IPsec Tunnels** and **Static Routes** settings.

NETWORK INTERFACES

[CONFIGURE > NETWORK CONNECTIONS > Network Interfaces](#)

23.10.3	CONFIGURE Menu	123
---------	----------------	-----



The interface supports both IPv4 and IPv6 networks. The IP address of the unit can be setup for Static or DHCP. The following settings can be configured for network ports:

- IPv4, IPv6
- Static and/or DHCP
- Enabling or disabling network interfaces
- Ethernet Media types

For detailed information about Network Interface configuration and adding a new connection, see ["Change Network Settings" on page 37](#).

For information about VLAN interfaces, bridges and bonds, see ["Network Aggregates - Bonds and Bridges" on page 126](#)

DNS CONFIGURATION

DNS settings such as Name Servers and Search Domains can be configured for each network interface, which will override the DHCP provided settings.

Name servers allow the system to resolve hostnames to IP addresses to communicate with remote systems. Search domains allow the system to resolve partially qualified domain names (PQDN) by appending entries from the listed search domains to form a fully qualified domain name (FQDN).

When adding an interface to a Bond or Bridge, it will use the DNS configuration of the aggregate interface.

Note: Interfaces must have at least one network connection to be able to perform DNS resolution.


CONFIGURE DNS VIA THE WEB UI




[CONFIGURE](#) > [NETWORK CONNECTIONS](#) > [Network Interfaces](#)


On the Network Interfaces page, select the desired interface and click the Edit link.

NAME SERVERS

1. Add one or more name servers to the list by clicking the **Add Name Server** button.
2. Name servers can be IPv4 or IPv6 addresses.
3. Name servers can be removed from the list by clicking the **Delete** button next to each row.
4. Click **Apply** to save the changes.


Name Server 




10.23.66.123	
10.23.66.124	
fd07:2218:1350:49::1:1531	


 Add Name Server

DNS SEARCH DOMAINS

1. Add one or more DNS search domains to the list by clicking the **Add Search Domain** button.
2. Search domains should be fully qualified domain names.
3. Search domains can be removed from the list by clicking the **Delete**

Search Domain 

office.example.com	
sales.example.com	
development.example.com	

 Add Search Domain

button next to each row.

4. Click **Apply** to save the changes.

CONFIGURE DNS VIA THE COMMAND LINE

Description	Command
Display configured DNS settings for an interface	<pre>ogcli get physif "net1"</pre>
Update DNS settings for an interface	<pre>ogcli update physif "net1" << END dns.nameservers[0]="1.1.1.1" dns.nameservers[1]="1.0.0.1" dns.search_domains[0]="example.net" dns.search_domains[1]="example.com" END</pre>
Check unbound service status	<pre>systemctl status unbound.service</pre>
List forward-zones in use	<pre>unbound-control list_forwards</pre>

NETWORK AGGREGATES - BONDS AND BRIDGES

[CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface](#)

23.10.3	CONFIGURE Menu	126
---------	----------------	-----

BRIDGES


Network bridges allow connecting of multiple network segments together so that they may communicate as a single network.

Definitions of the bridge details as in the **Bridge Form Definitions** table later in this topic.

Note: Whether creating a new bridge or editing an existing bridge the page is very similar.

CREATE A NEW BRIDGE

To create a new bridge:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the **New Bridge**  button that is located at the top-right of the window.
3. Select which interface will serve as the primary interface for the new bridge.

Note: When the primary interface is selected, its MAC address is displayed in the MAC address field. This MAC address is inherited by the new bridge interface.

4. Complete the new bridge details form as in the **Bridge Form Definitions** table.
5. Click the **Create** button to finalize the creation of the new bridge.

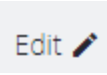
EDIT AN EXISTING BRIDGE

To edit an existing bridge:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the bridge that you would like to edit, the bridge details are expanded.
3. Click on the bridge **Edit** button that is located next to the Enable / Disable toggle buttons.
4. Select which interface will serve as the primary interface for the new bridge.
5. Change the bridge details as required in accordance with the **Bridge Form Definitions** table.
6. Click the **Update** button to finalize the edit process. Updating the bridge will temporarily interrupt network activity on this interface.

Note:Editing the primary interface will not update its connections.

EDIT BRIDGE - FORM DEFINITIONS

New Bridge Field	Definition
Description	The editable Description field allows you to add a description of the interface. If the description field is not completed the field will default to a computed value to describe the interface.
Enable Spanning Tree Protocol	Enable or disable Spanning Tree Protocol. See "Spanning Tree Protocol" on page 134.
Network Interface Selection	Click the checkbox of each network interface you want to include in the bridge. Available interfaces include Ethernet and VLAN interfaces that are not part of another bond or bridge. Bond interfaces can be included in a bridge by using the ogcli tool. See Support for Bonds in Bridges on Zendesk.
Primary Interface	Select the interface that is to be used for selecting the MAC address of the aggregate. The new bond inherits the MAC address of the primary interface. On creation, any Network Connections which exist on the Primary Interface will be attached to the Bond/Bridge after it is initially created. When a Bond/Bridge is deleted, any Network Connections which exist on the aggregate interface are handed over to the Primary Interface.
Inherited Connections	When the Primary Interface is selected, the connections inherited by the new bridge are listed here.
	Click to edit the details of an existing interface.

BONDS

Network bonds allow combining two or more network interfaces together into a single logical "bonded" interface for load balancing, redundancy or improved performance depending on the bond mode used.

Definitions of the bond details as in the **Bond Form Definitions** table later in this topic.

Note: Whether creating a new bond or editing an existing bond the page is very similar.

CREATE A NEW BOND

To create a new bond:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the **New Bond** button that is located at the top-right of the window.
3. Select which interface will serve as the primary interface for the new bond.

Note: When the primary interface is selected, its MAC address is displayed in the MAC address field. This MAC address is inherited by the new bond interface.

4. Complete the new bond details form as in the **Bond Form Definitions** table.
5. Click the **Create** button to finalize the creation of the new bond. Network connections from non-primary interfaces will be deleted when the new bond is created.

EDIT AN EXISTING BOND

To edit an existing bond:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the bond that you would like to edit, the bond details are expanded.
3. Click on the bond **Edit** button that is located next to the Enable / Disable toggle buttons.
4. Change the bond details as required in accordance with the **Edit Bond Form Definitions** table below.
5. Click the **Update** button to finalize the edit process. Updating the bond will temporarily interrupt network activity on this interface.

Note:Editing the primary interface will not update its connections.

EDIT BOND - FORM DEFINITIONS

New Bond Field	Definition
Description	<p>The editable Description field allows you to add a description of the interface. If the description field is not completed the field will default to a computed value to describe the interface.</p>
Mode	<p>The mode determines the way in which traffic sent out via the bonded interface is dispersed over the real interfaces. Available modes are:</p>
	<p>Round Robin Balancing - Packets are sequentially transmitted/received through each interface, one by one.</p>
	<p>Active Backup - If the active secondary interface is changed during a failover, the bond interface's MAC address is then changed to match the new active secondary's MAC address.</p>
	<p>XOR Balancing - Balances traffic by splitting up outgoing packets between the Ethernet interfaces, using the same one for each specific destination when possible.</p>
	<p>Broadcast - All network transmissions are sent on all secondary interfaces. This mode provides fault tolerance.</p>
	<p>802.3ad (Dynamic Link Aggregation) - Aggregated NICs act as one NIC, but also provides failover in the case that a NIC fails. Dynamic Link Aggregation requires a switch that supports IEEE 802.3ad.</p>
	<p>Transmit Load Balancing - Outgoing traffic is distributed</p>

	<p>depending on the current load on each secondary interface. Incoming traffic is received by the current secondary interface. If the receiving secondary fails, another secondary takes over the MAC address of the failed secondary.</p> <p>Adaptive Load Balancing - Includes transmit load balancing (tlb) and receive load balancing (rlb) for IPv4 traffic and does not require any special switch support.</p>
Poll Interval	<p>The poll interval specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each secondary is inspected for link failures. A value of zero will disable MII link monitoring.</p>
Network Interface Selection	<p>Click the checkbox of each network interface you want to include in the bridge.</p> <p>Available interfaces include Ethernet and VLAN interfaces that are not part of another bond or bridge.</p>
Primary Interface	<p>Select the interface that is to be used for selecting the MAC address of the aggregate. The new bond inherits the MAC address of the primary interface. On creation, any Network Connections which exist on the Primary Interface will be attached to the Bond/Bridge after it is initially created. When a Bond/Bridge is deleted, any Network Connections which exist on the aggregate interface are handed over to the Primary Interface.</p>
Active Connections	<p>When the Primary Interface is created, the connections inherited by the new bond are listed here. When edited, Active Connections on the aggregate will not be updated if the primary interface is changed.</p>



Click to edit the details of an existing interface. Updating a bridge will temporarily interrupt network activity on the interface when you click the **Update** button.

SPANNING TREE PROTOCOL

[CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface](#)

Spanning Tree Protocol (STP) allows an Console Manager to discover and eliminate loops in network bridge links, preventing broadcast radiation and allowing redundancy.

When STP is implemented on switches to monitor the network topology, every link between switches, and in particular redundant links, are cataloged. The spanning-tree algorithm blocks forwarding on redundant links by setting up one preferred link between switches in the LAN. This preferred link is used for all Ethernet frames unless it fails, in which case a non-preferred redundant link is enabled.

Note:STP Limitations

If multiple bridges are created on the same switch, they should not be used on the same network segment as they have the same MAC addresses; therefore, STP will likely not work correctly as they will have the same bridge id.

Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP) and other proprietary protocols are not supported.

The bridge settings relating to STP cannot be changed from the default values shown below:

group_address

forward_delay (default is 15)

hello_time (default is 2)

max_age (default is 20)

priority (default is 32768 (0x8000))

ENABLE STP IN A BRIDGE

To enable STP you can use the UI or CLI. The procedures are:

BRIDGE WITH STP ENABLED - UI

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface > New Bridge page

1. In the **Network Interfaces** page, click the **Create New Bridge** button.
2. Click to select the **Enable Spanning Tree Protocol** option.



NEW BRIDGE

Description

New Bridge

Enable Spanning Tree Protocol ?

BRIDGE WITH STP ENABLED - OGCLI

```
admin@cm8148:~# ogcli get physif system_net_physifs-5
  bridge_setting.id="system_net_physifs-5"
bridge_setting.stp_enabled=true
description="Bridge"
device="br0"
enabled=true
id="system_net_physifs-5"
media="bridge"
name="init_br0"
  slaves[0]="net2.3"
```

BRIDGE WITH STP DISABLED - OGCLI

```
admin@cm8148:~# ogcli update physif system_net_physifs-5 bridge_
setting.stp_enabled=false
  bridge_setting.id="system_net_physifs-5"
  bridge_setting.stp_enabled=false
description="Bridge"
device="br0"
  enabled=true
id="system_net_physifs-5"
media="bridge"
name="init_br0"
  slaves[0]="net2.3"
```

IPSEC TUNNELS

CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels

The Opengear Console Manager (CM) can use IPsec to securely connect and route between two or more LANs (sometimes referred to as site to site, LAN-to-LAN, L2L VPN), or as a single client endpoint connecting to a central LAN or endpoint (sometimes referred to as host to site, or host to host).

IPsec does not make a formal distinction between initiator and responder, however the Opengear CM can both initiate tunnels (as the "initiator") and have other devices initiate tunnels to it (as a "responder").

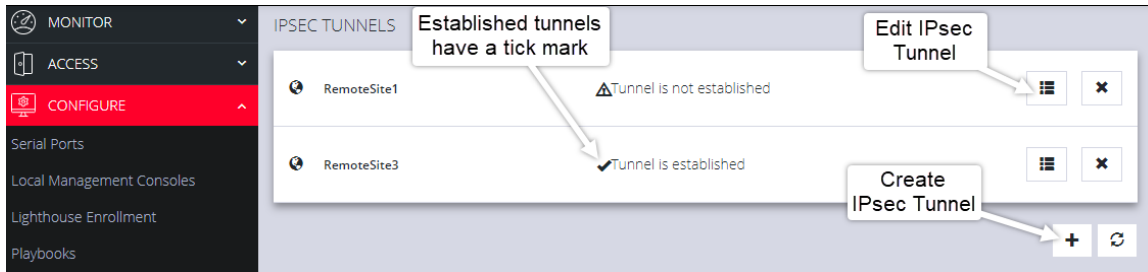
CREATE, ADD OR EDIT IPSEC TUNNELS

On the IPsec Tunnels page, you can create, edit, and delete IPsec tunnels.

To create an IPsec tunnel:

1. Click **CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels**.

The IPsec Tunnels page with two tunnels previously created.



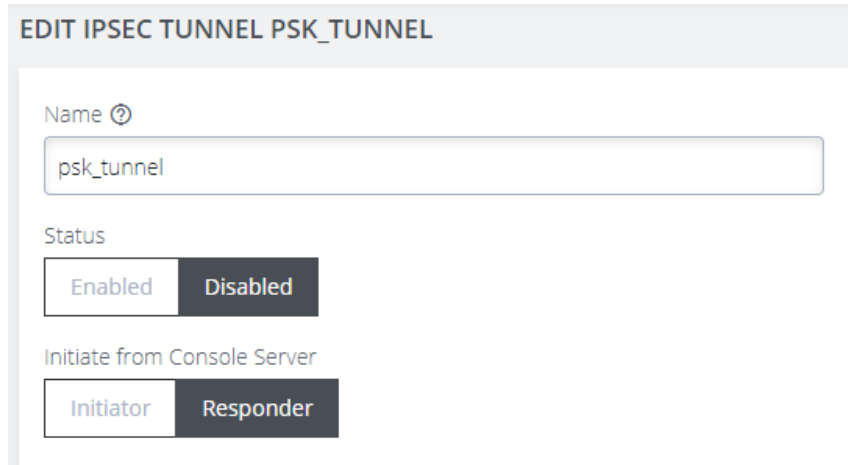
*If there are no existing tunnels, this **Create Tunnel** button is displayed:*



2. Click **CREATE TUNNEL**. This opens the **EDIT IPSEC TUNNEL** page.

NAME and STATUS

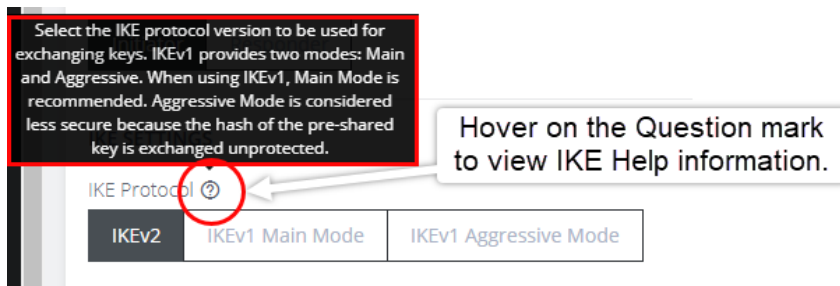
3. In the **Name** section of the page, give your new tunnel a unique name and click the **Enabled** button.



4. Set the Console Server to be the **Initiator** or **Responder**.

Note:When **Initiator** is selected, the node will actively initiate the tunnel by sending IKE negotiation packets to the remote end.

IKE SETTINGS



Continued...

5. Select an **IKE Protocol** version to use for exchanging keys. IKEv1 provides two modes: **Main** and **Aggressive**. When using IKEv1, Main Mode is recommended. Aggressive Mode is considered less secure because the hash of the pre-shared key is exchanged unprotected.
6. Select the **Algorithm Proposal**. This is a set of algorithms used for negotiation when attempting to establish the IPsec tunnel. By default, the node will attempt to negotiate the tunnel using a list of common algorithms which are considered safe. Alternatively, a set of default proposals that guarantee Perfect Forward Secrecy (PFS) can be selected.
7. Select **Initiate** to actively initiate the tunnel by sending IKE negotiation packets to the remote end.
8. Set up the **Phase 1** and **Phase 2** time interval between the key material refresh of the IKE and Child.

AUTHENTICATION

CM Authentication can use PSK or PKI.

9. **For pre-shared key (PSK) authentication**, enter a pre-shared secret key; both ends of the tunnel must use the same key.

Tip:

To construct ID_USER_FQDN identities, use `user@example.com`

To construct ID_FQDN type identities, use `@host.example.com`

If left blank, the outer local IP address of the tunnel is used as the identity.

10. Enter a **Local ID** Identity or IP address for the local end of the tunnel. If left blank, the outer-local IP address is used as the source address of the tunnel.
11. **For Public Key Infrastructure (PKI) authentication**, upload the certification bundle file or, drag and drop the file into the Certificate Bundle field.



TUNNEL SETTINGS

12. Select **Enabled** if enforced UDP encapsulation is required. When enabled, the IKE daemon can simulate the NAT detection payload.

ADDRESSING

13. Enter the **Local Address** to be used as the source address of the tunnel. If left blank, IPsec will automatically use a default.
14. Enter a **Local Subnet**. Specify local traffic to be tunneled. When no subnets are specified, only traffic originating from this device will be tunneled.
15. Enter the **Remote Address** or hostname for the remote end of the tunnel. If left blank, IPsec will accept initiation packets from any address.
16. Enter the **Remote Subnet**. Specify addresses or subnets that are behind the remote end of this tunnel. If no subnet is specified, only traffic originating from the outer remote address will be accepted.

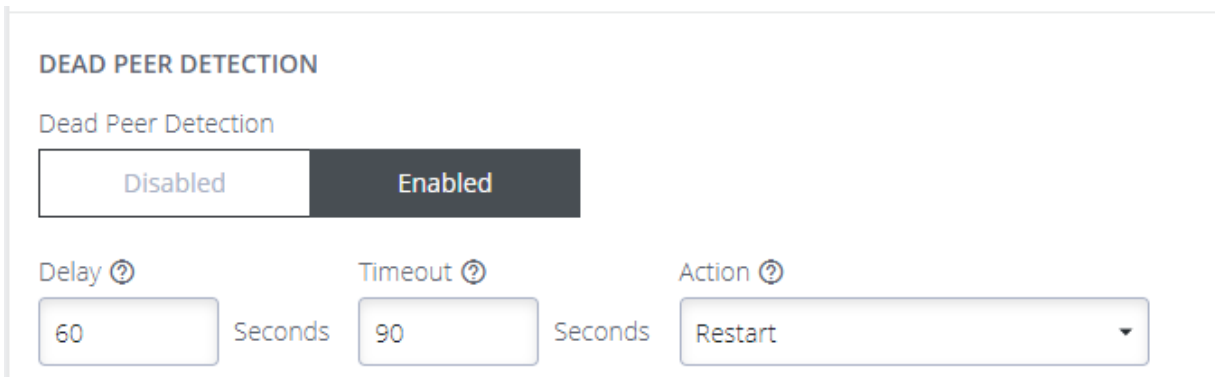
DEAD PEER DETECTION

Tip: Dead Peer Detection may be used to support long-lived tunnels.

Dead Peer Detection (DPD) is a method used by nodes to verify the current existence and availability of IPsec peers. A node performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgments (R-U-THERE-ACK messages) from the peer.

Continued...

You can enable DPD and configure the various options to fine-tune the functionality:



- **Delay** - the time interval between polling the peer (default is 60 seconds).
- **Timeout** - the waiting time before deciding that a peer connection is not live (default is 90 seconds).
- **Action** - the action to be performed when a connection is timed-out. (default is Restart).
 - **Restart** will immediately attempt to renegotiate the tunnel.
 - **Clear** will close the CHILD_SA.
 - **Trap** will catch matching traffic.

ENABLE the IPsec TUNNEL

17. When you have completed the IPsec Tunnel set-up process, ensure the IPsec tunnel status is set to **Enabled**, then, click **Save**.

The new tunnel is now listed on the **CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels** page.

STATIC ROUTES

[CONFIGURE > NETWORK CONNECTIONS > Static Routes](#)

23.10.3	CONFIGURE Menu	142
---------	----------------	-----

Static routes are predefined paths that traffic can be configured to take through the network for purposes such as security, cost or to override the default route.

The list of configured static routes are displayed in a table with their current status indicated by the status column.

Status	Meaning
Installed	The route is installed in the routing table.
Not Installed	The route may not be currently installed, but should update in a moment.
Error	The route failed to be installed.
Failed to fetch status	There is an error with the system and status failed to be obtained. This is a temporary error and should update in a moment.
The network interface is disabled	The static route is bound to an interface which is not enabled.
The network interface is disconnected	The static route is bound to an interface which is not connected.
The network	The route cannot be installed as there are no active connections on this interface.

Status	Meaning
interface has no active connections	

CONFIGURE STATIC ROUTES

On the Static Routes page you can add, edit or delete static routes.

Note: Only basic validation is performed when static routes are saved. Check the status column to ensure your route is installed and working correctly.

CREATE A STATIC ROUTE

1. Click the **Add** button to navigate to the creation page.
2. Enter a valid IPv4 or IPv6 destination address or network, followed by the netmask in CIDR notation. The destination address/network must be unique.
3. Enter the gateway or select an interface for the static route to use.
4. Optionally, provide a metric for the route. Routes with a lower metric value are higher priority.

Destination Address	Default Metric
IPv4	0

Destination Address	Default Metric
IPv6	1024

5. Click the **Apply** button to save the changes.
6. If the changes are saved successfully you are returned to the Static Routes list page.
 - If there is an error with the configuration and the route fails to install, a red banner is displayed.
 - If the route installed successfully, a green success banner is displayed.
7. The current status of the configured route is displayed in the table, which may change depending on the status of the network configuration.

EDIT A STATIC ROUTE

1. Click the description of the desired static route in the list to access the **Edit** page.
2. Update the details of the static route.
3. Click apply to save the changes.

DELETE A STATIC ROUTE

1. Click the description of the desired static route in the list to access the **Edit** page.
2. Click the **Delete** button at the top-right of the page.
3. Click **Yes** to confirm the action.

4. If the route was removed from the routing table as expected, a green success banner is displayed.

MANAGING STATIC ROUTES VIA COMMAND LINE

Administrative users can also view the status and perform configuration of static routes via the command line interface.

After creating or modifying a route via the command line, you should take note of the route id and confirm that it has been installed successfully in the routing table.

Description	Command
Display IPv4 installed routes	<pre>ip route</pre>
Display IPv6 installed routes	<pre>ip -6 route</pre>
Display all route information	<pre>ip route show table all</pre>
Show status of configured routes via ogcli	<pre>ogcli get monitor/static_routes/status</pre>
Get static route configuration via	<pre>ogcli get static_routes</pre>

Description	Command
ogcli	
Create static route via ogcli	<pre>ogcli create static_route << END destination_address="1.1.1.1" destination_netmask=32 gateway_address="1.1.1.1" interface="net1" metric=0 END</pre>
Update static route via ogcli	<pre>ogcli update static_route "1.1.1.1" << END interface="net2" metric=100 END</pre>
Delete static route via ogcli	<pre>ogcli delete static_route "1.1.1.1"</pre>

NETWORK RESILIENCE

[CONFIGURE > NETWORK RESILIENCE >](#)

Under the NETWORK RESILIENCE menu, you can manage Out-of-Band (OOB) settings.

23.10.3	CONFIGURE Menu	147
---------	----------------	-----

OUT OF BAND FAILOVER

CONFIGURE > NETWORK RESILIENCE > OOB Failover

Out-of-Band (OOB) Failover detects network disruption via the probe interface, and automatically activates a cellular or ethernet interface connection to re-establish network access.

OOB failover requires an IPv4 address (in dotted decimal format), or an IPv6 address, or a domain name, which is always reachable and unlikely to change. When OOB failover is **Enabled**, the node regularly pings this address, using the probe interface, to check for network connectivity.

When OOB Failover is **Enabled**, and the device enters the `failover_starting` state, the device will establish a connection on the `failover_physif` (enabling the `failover_physif` in the process, if it wasn't already enabled).

Note: It can take a while to transition between the `failover_starting` state and the `failover_complete` state. This transition is usually not more than a couple of seconds for wired connections. Cellular connections can take a few minutes to establish, however. If the chosen `failover_physif` was enabled in the Web UI at **Configure > Network Connections > Network Interfaces** and already had a connection established, this transition will be faster than if the `failover_physif` was disabled.

When in the `failover_complete` state, the device will continue to perform connectivity tests against the configured probe addresses from the `probe_`



interface. When connectivity is restored, the `failover_physif` will return to the enabled/disabled status it had before it was used for a failover connection, and the device will transition to the `primary_complete` state.

OPTIONAL ADDITIONAL PROBE ADDRESS

You can, if preferred, configure an optional, (secondary) additional probe address (`probe_address_2`) for the connectivity tests associated with Out of Band Failover. When the additional probe address (`probe_address_2`) is configured, the device will only activate the `failover_starting` state change when both primary and additional probe addresses are unreachable. When an additional probe address is not specified (empty), the connectivity tests will only check against the `probe_address`, and enter the `failover_starting` state when it is unreachable.

SHOW OOB FAILOVER SETTINGS - CLI CONFIGURATION EXAMPLE

```
root@om2216-1:~# config
Welcome to the Opengear interactive config shell. Type ? or help for help.
config: failover/settings
config(failover/settings): show
Entity failover/settings
dormant_dns      false
enabled          false
failover_physif ""
probe_address   8.8.8.8
probe_address_2 ""
probe_physif     ""
config(failover/settings): enabled true
config(failover/settings): failover_physif wwan0
config(failover/settings): probe_physif net1
config(failover/settings): probe_address_2 1.1.1.1
config(failover/settings): apply
Updating entity failover/settings.
config(failover/settings): show
Entity failover/settings
dormant_dns      false
enabled          true
failover_physif wwan0
probe_address   8.8.8.8
probe_address_2 1.1.1.1
probe_physif     net1
config(failover/settings): █
```

ENABLE OUT-OF-BAND FAILOVER

1. To manage Out-of-Band Failover, navigate to the **CONFIGURE > NETWORK RESILIENCE > OOB Failover** page.

OOB FAILOVER

FAILOVER SETTINGS

Status

Enabled Disabled

Probe Interface ?

NET2 - 1G Copper/SFP

i Once connectivity to configured probe addresses is lost, failover can take up to 5 minutes to activate.

Probe address ?

8.8.8.8

Additional probe address ?

1.0.0.1

Probe Interface: this is the interface that will be used to test if ping can reach the configured address

Probe Address: the ipv4 or ipv6 or domain name of the address that will be “pinged”.

Additional Probe Address: the ipv4 or ipv6 or domain name of the additional, secondary probe address that will be “pinged” if the first probe address is unreachable.

2. In the **Failover Interface** section, select the failover interface from the drop-down list.



Configurable probe (failover from) and failover (failover to) interfaces are shown below:

NET1 - the default probe interface.

Cellular - the default failover interface for cellular-capable models.

NET2 - the default failover interface for non-cellular models.

3. When you have completed the OOB Failover set-up, ensure the OOB Failover status is set to **Enabled**, then, click **Apply**, a confirmation is displayed.
4. On the **Network Interfaces** page the Failover Interface will display "Configured for OOB Failover" beside the interface name.



5. When failover is triggered, the interface will be marked with the warning: **OOB Failover Active** to an Admin user when logged in.



Note: It may take up to five minutes for a failover to actually occur once the probe stops connecting to the probe address.

Note:The shortcut button **Enabled/Disabled** is disabled or removed when an interface is in active failover.

DNS QUERIES ON A DORMANT FAILOVER INTERFACE

The Dormant DNS option allows DNS queries on the failover interface to be disabled in normal operation so that DNS queries can be paused.



The option configures how the DNS name servers and search domains configured for the failover interface are used by the system.

- If set to **Yes**, the DNS name servers and search domains configured for the failover interface will always be available to the system for DNS name resolution. Allowing DNS queries while failover has not been triggered make it more likely that DNS requests will be made over the cellular interface which will increase data usage.
- If set to **No**, the DNS name servers and search domains will be made available to the system only when the failover state is active.

To configure the DNS name servers and search domains, see "[DNS Configuration](#)" on page 124.

OOB FAILOVER TYPES & FAILOVER BEHAVIOR

OOB Setting	Failover Interface	Mode	Description
Disabled	Enabled	Always up OOB	<p>When OOB Failover is disabled, the default outgoing interface cannot be specified, the default route is selected automatically.</p> <p>Outbound network connections (e.g. VPN client tunnels, SNMP alerts) are established according to the main static routing table, regardless of network state.</p>
Enabled	Disabled	Failover mode	<p>Failover detection is enabled on the selected “probe” interface. The network or cellular interface remains in a down state with no network configuration.</p> <p>When failover is initiated, the network or cellular interface is started and configured. If a default route is installed on the interface, it takes precedence over the default route on the failed “probe” interface. Outbound network traffic (e.g. VPN client</p>

			<p>tunnels, SNMP alerts) are established or re-established over network or cellular connection during failover.</p> <p>The advantage of this mode is the secondary connection is completely inactive during normal operation which may be advantageous where the goal is to keep the interface off the Internet as much as possible, e.g. a cellular plan with expensive data rates and no carrier-grade NAT.</p>
Enabled	Enabled	Dormant failover	<p>Failover detection is enabled. Only inbound connections on the network or cellular interface are routed back out the network or cellular interface, to enable OOB access from remote networks (e.g. incoming SSH). Otherwise, outbound network connections (e.g. VPN client tunnels, SNMP alerts) are established according to the main static routing table, regardless of network state.</p> <p>When failover is initiated, the default route of the network or cellular interface takes pre-</p>

			<p>cedence over the failed “probe” interface. Out-bound network traffic (e.g. VPN client tunnels, SNMP alerts) are established or re-established over the network or cellular connection during failover.</p> <p>The advantage of this mode is the network or cellular connection is available for inbound out-of-band access during normal operation.</p>
--	--	--	--

USER MANAGEMENT

CONFIGURE > USER MANAGEMENT

Under the User Management menu, you can create, edit, and delete groups and users, as well as assign users to groups. You can also set up remote user authentication.

GROUPS

CONFIGURE > USER MANAGEMENT > Groups

Groups are used to grant privileges to users. When a user is a member of a group, defined privileges may be granted to the group by an administrator. When editing a group, the (authorised) user selects from a list of devices, all of which are under the heading **SERIALLY CONNECTED DEVICES**.

23.10.3	CONFIGURE Menu	156
---------	----------------	-----

PERMISSION CHANGES IN THE WEB UI

A new feature change called Access Rights is introduced in release 22.11 replaces the previous concept of a user *Role* and instead uses a set of configurable *Access Rights* for each group. Each access right governs access to a particular feature (or set of highly related features), with a user only having access to features for which they have an assigned access right.

Tip: To support the new permissions model several rest API endpoints have been updated for the new functionality. Wherever possible, these changes backwards compatible. See the release noted for details.

UNDERSTANDING ACCESS RIGHTS

An access right is a permit authorizing access to a feature or collection of related features. Holders of the permit (i.e. the access right) are given access to the feature.

A user gains access rights by the following:

- Access Rights are assigned to Groups.
- Users are members of zero or more Groups.
- A User inherits all Access Rights from all the Groups they are a member of.

Some features may require the user to hold multiple access rights to access the feature through a specific interface. For example, a user needs the “right to use the web UI” and the “right to configure serial ports” to make configuration changes to a serial port through the web UI.

DEFINED ACCESS RIGHTS

23.10.3	CONFIGURE Menu	157
---------	----------------	-----

There are four *defined* rights (admin, web_ui, pmshell, and port_config) as summarized in the following table.

Access Rights	Description
admin	The admin access right grants a holder access to everything; every feature and every user interface.
web_ui	Permits access for an authenticated user to basic status information via the web interface and rest API. Users can: <ul style="list-style-type: none">• Make requests to the subset of endpoints that provide this same information. In both cases the user must be authenticated.• See information about their own user and groups.• See serial port status information for the specific ports the user is granted access to.
pmshell Restricted CLI	Permits access to devices connected to serial ports. Does not give permission to configure all serial ports, only to those that are added to the same group containing the pmshell rights.
Port Config	Permits access to configure serial ports. This access right gives the holder the ability to configure serial ports. This right does not give the holder the ability to access the serial port.

Tip: A right may be combined with another right for a feature to be accessible by a user. For example, `web_ui` to login and `port_config` to configure a serial port. The `port_config` right by itself is not useful.

Admin Access Right (`admin`)

Any user who was previously an Administrator role now inherits the `admin` access right, giving that user the same “can do everything” permission.

Tip: The **Admin Access** toggle switch in the Web UI hides other rights selections as Admin Access overrides all other rights.

Web UI Access Right (`web_ui`)

Any user who was previously a Console User role now inherits the `web_ui` and `pmshell` access rights and there are no functional changes for this user.

Tip: From release 22.11 in the Web UI, the **Rights** checkbox replaces the **Roles** drop-down selection.

The `web_ui` access right grants the user the ability to

- log into the Web UI,
- see a listing of serial ports (The “Access → Serial Ports” menu item) and to
- edit a restricted set of user configuration such as changing their own password.

Portmanager Shell Access Right (`pmshell`)

Any user who was previously a Console User role now inherits the `pmshell` access rights and there are no functional changes for this user.



The `pmshell` access right grants the user access to the serial port web terminals and the ability to use `pmshell` over SSH. These rights are applied only to the access ports to which they have been granted rights.














Port Configuration Access Right

The `port_config` access right grants the holder of this right the ability to make configuration changes to the serial ports they have been assigned. Note that a user without the `web_ui` right cannot login to the web UI to configure serial ports, so a user must inherit the `web_ui` from at least one group.

Access > Serial Ports View

Users with the `port_config` access right to some serial ports are able to see the **Edit** link on the **Access > Serial Ports** page for those ports only. Non-admin users with the `port_config` role are able to see any active sessions on a port, but are not able to terminate the session.

LOGGING LEVEL	ESCAPE CHARACTER
Logging Disabled	~

 Port-3 Port-3, 9600-8-N-1-X2	 Console Server	 0 Sessions	 
 Port-4 Port-4, 9600-8-N-1-X2	 Console Server	 0 Sessions	 
 Port-5 Port-5, 9600-8-N-1-X2	 Console Server	 0 Sessions	

Configure > Serial Ports View

The Configure Serial Ports page is accessible to users with the `port_config` and `web_ui` access rights appear in the navigation sidebar menu. This page lists ports that the user has both `port_config` and `web_ui` access rights.

23.10.3	CONFIGURE Menu	160
---------	----------------	-----

Tip: It is possible to edit all details on these ports, however, changing the “mode” of a port will disconnect any sessions.

Non-Admin Users

Non-admin users with `port_config` access right are able to perform Serial Port Autodiscovery on the ports that they are able to configure. If autodiscovery is already running, they will be able to see the banner but will not be able to view the autodiscovery logs or cancel the running job. Non-admin users are not able to configure the Serial Port Autodiscovery Schedule and the icon is hidden, but are able to see which ports are configured of the ports to which they have access.

PROTECTED GROUPS AND USERS

Certain types of groups and users have protected status, meaning that they cannot be changed or deleted. Protected groups comprise the following:

`root` - The root user is hard-coded member of the admin group. As such, the root user cannot be deleted.

`admin` - The admin group cannot be disabled or changed to a non-admin group.

`netgrp` - The special ‘netgrp’ also cannot be deleted. This group is assigned to users from AAA auth that don’t have a group assigned from the authentication server.

Tip: For these protected groups no 'Delete' button appears beside them in the Web UI.

UNDERSTANDING SERIAL PORT ACCESS

Serial ports are assigned to a group in the same way as access rights are assigned to a group, however, it is the access rights that are assigned to the same group that determine what a user can actually do with those serial ports. The access rights assigned to one group will only apply to the serial ports assigned to that same group, they do not apply to the serial ports of another group.

For example, a user in a group with `port_config` and `port-01` can configure that port but not access the device (as that requires `pmshell` access rights).

Consider the following two groups, *Accounts Admin* and *Port #03 User*:

Group Name	Accounts Admin	Port #03 User
Access Rights	<code>port_config</code> <code>web_ui</code>	<code>pmshell</code> <code>web_ui</code>
Serial Ports	<code>port-01</code> <code>port-02</code>	<code>port-03</code>

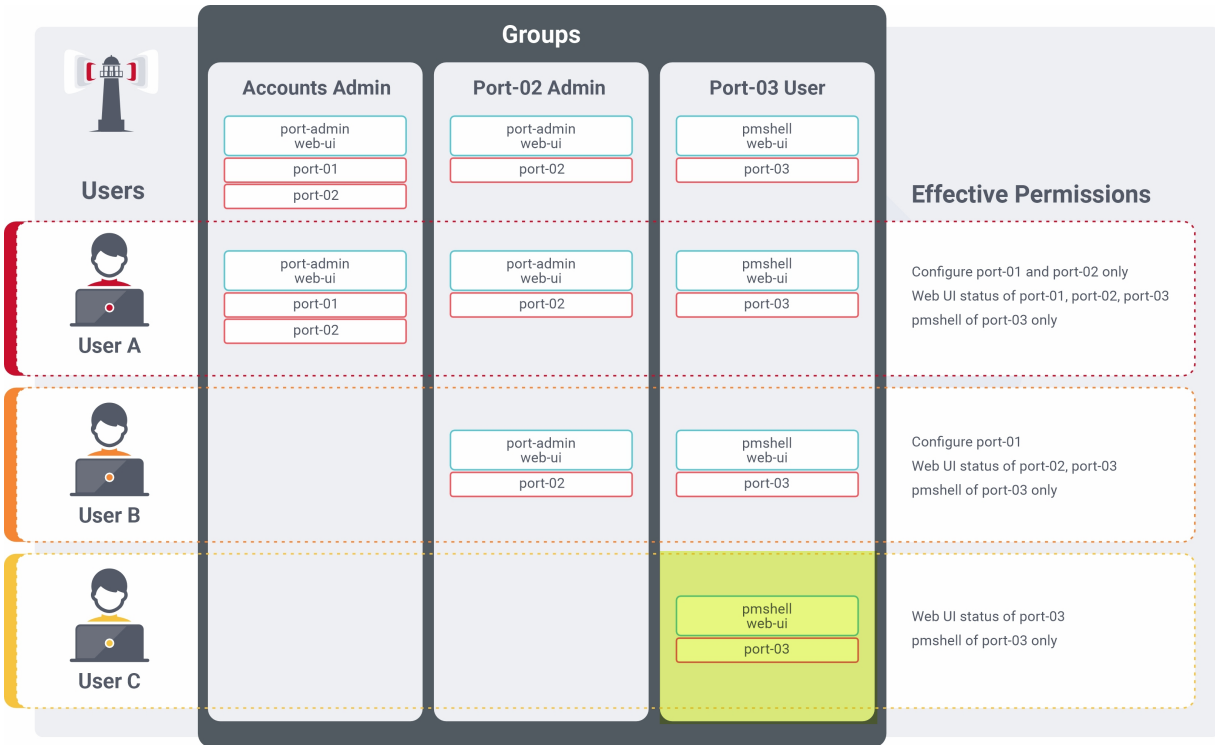
The effective rights for a user in one or both of those groups is shown in the following table. It shows how access rights assigned to one group will only apply to the serial ports assigned to that same group:

The following table shows the effective rights for a user in one or both of those groups, *Accounts Admin* and *Port #03 User*:

Group Membership	<i>Accounts Admin</i>	<i>Port #03 User</i>	<i>Accounts Admin & Port #03 User</i>
Action			
Configure port-01	✓	✗	✓
Configure port-02	✓	✗	✓
Configure port-03	✗	✗	✗
Access port-01	✗	✗	✗
Access port-02	✗	✗	✗
Access port-03	✗	✓	✓

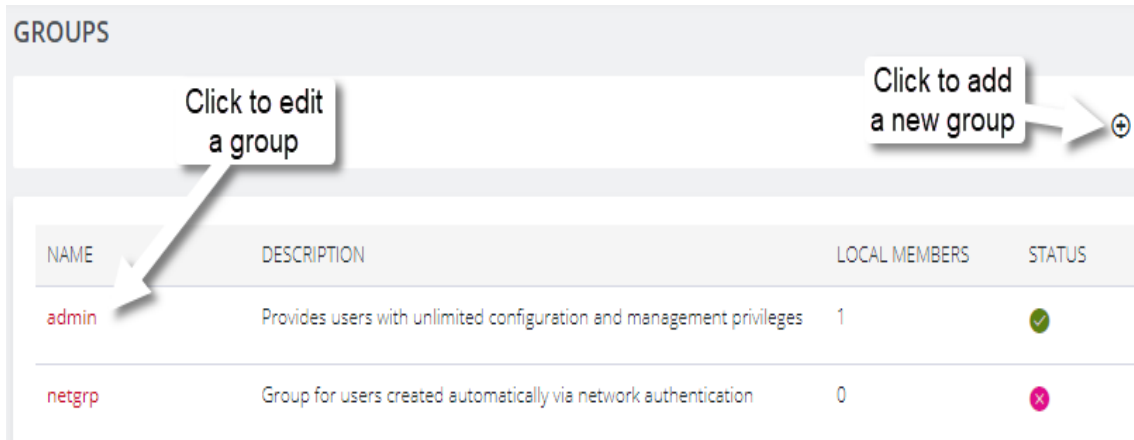
Note:Note the highlighted cell; a user with `pmshell` access to `port-03` (from the *Port #03* user group) does not also get `port_config` for that port, even though that access right is inherited from the *Accounts Admin* group. The access rights of a group *only apply to the serial ports in that same group*. This principle is illustrated in the following figure:

The figure below shows how access rights assigned to one group only apply to the serial ports assigned to that same group.






CREATE A NEW GROUP

1. Select **CONFIGURE > USER MANAGEMENT > Groups**.



NAME	DESCRIPTION	LOCAL MEMBERS	STATUS
admin	Provides users with unlimited configuration and management privileges	1	✔
netgrp	Group for users created automatically via network authentication	0	✘

	Add a new group.
admin	Click on the group name to edit an existing group.
Status <input type="checkbox"/> Enabled <input checked="" type="checkbox"/> Disabled	In the EDIT GROUP window - Enable/Disable an existing group.
Admin Access  <input checked="" type="checkbox"/> Enabled <input type="checkbox"/> Disabled	Grant admin access rights and full control of this console, and all attached devices, to all users of this group.
	Delete a group (or delete selected groups).

2. Click the **Add New Group** button. The **CREATE GROUP** page opens.

CREATE GROUP

Status
 Enabled Disabled

Name ⓘ

Description ⓘ

Admin Access ⓘ
 Enabled Disabled

ACCESS RIGHTS

NAME	DESCRIPTION
<input type="checkbox"/> Web UI	Permits access for an authenticated user to basic status information via the web interface and rest API.
<input type="checkbox"/> PM Shell (Restricted CLI)	Permits access to devices connected to serial ports.
<input type="checkbox"/> Missing translation: general.access_rights.rights.port_config	Missing translation: general.access_rights.rights.port_config.description

3. Enter a **Group Name**, **Description**, and set **Admin Access** to **Enabled** or **Disabled**. Specific access rights can be selected in the **ACCESS RIGHTS** area.

Note: **Group Name** is case sensitive. It can contain numbers and some alphanumeric characters. When using remote authentication, characters from a user's remote groups that are not allowed are converted to underscores during authentication. Local groups can be created that take that into account, allowing the authentication to continue.

Note: If **Admin Access** is Enabled, members of the group will have full access to and control of selected managed devices, and the rights that are selected under **ACCESS RIGHTS** for that group.

4. Select the applicable **Access Rights** for the group (see the below table).
5. If the new group is to be activated immediately, set the group Status to **Enabled**.

6. Click the **Submit** button to save the group. After creation, group **Status** and **Admin Access** may be enabled or disabled from the **CONFIGURE > USER MANAGEMENT > Groups > EDIT GROUP** page.

EDIT AN EXISTING GROUP

1. Select **CONFIGURE > USER MANAGEMENT > Groups**.
2. Click on the name of the group to be modified and make desired changes.
3. Click **Submit** to save the changes

The **CONFIGURE > User Management > Groups** page also allows administrators to delete a group. Users who were members of the deleted group lose any access and administrative rights inherited from the group.

Note:The netgrp group is inherited as the primary group for all remote AAA users who are not defined locally. By default, netgrp has the Administrator role and is disabled. It must be enabled to take effect for remote AAA users.

Note:For users that don't have any group, they are still part of netgrp, even if the netgrp membership is not explicitly enabled for the user.

The permissions for the netgrp members is a union of the permissions that have been given in the netgrp AND the permission for the user in AAA (TACACS+, RADIUS, etc).







If your netgrp "role" says "Console User" and you have priv-lvl 13 in TACACS+ (level 15 being the highest), then the union of that is like an admin already, so setting "console user" in netgrp does not matter.

LOCAL USERS

[CONFIGURE > USER MANAGEMENT > Local Users](#)

The Local Users feature allows a single point for the creation or management of local user accounts. The Local Users feature can use SSH authorized keys to control user access by using their local password; it is a point of control for:







- Authentication and authorization.
- Creating and editing user descriptions.
- Local passwords.
- User roles (admin or co sole user).
- Accessible ports.

LOCAL USERS			
			
<input type="checkbox"/>	Username	Description	Actions
<input type="checkbox"/>	root	System wide SuperUser account	  

See the Button Action Definitions table on the following page:

23.10.3	CONFIGURE Menu	168
---------	----------------	-----

Button Action Definitions:

	Add a new local user.
	Edit an existing user.
	Enable an existing user.
	Manage SSH Authorized Keys.
	Disable an existing user (or disable selected users).
	Delete a user (or delete selected users).


CREATE A NEW USER WITH PASSWORD

Note:Users are prevented from using the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This password policy applies to the WebUI, Config Shell and CLI. users configured on the system using software versions prior to 23.10 with password “default” are forced to change the user password to something other than “default” after upgrading to 23.10. This password feature update applies to configured boxes with existing users, not just factory defaulted software.

1. Navigate to the **CONFIGURE > USER MANAGEMENT > Local Users** page.

23.10.3	CONFIGURE Menu	169
---------	----------------	-----



2. Click the **Add User**  button. The **New User** dialog appears.
3. Enter a Username, Description, and Password that the new user will use.
4. Re-enter the **Password** in the **Confirm Password** field.
5. Select the **Enabled** checkbox.
6. Click **Apply**. A banner will confirm that the data has been saved.

CREATE A NEW USER WITH NO PASSWORD (REMOTE AUTHENTICATION)

To create a new user with no password.

Note: If a new user is created with no password, this will cause the user to fall-back use remote authentication.

1. Select **CONFIGURE > User Management > Remote Authentication**
2. Select a Mode.
3. Enter Settings and click **Apply**.
4. Select **CONFIGURE > USER MANAGEMENT > Local Users**
5. Click the **Add User** button. The **New User** dialog loads.
6. Enter a **Username**, **Description**.
7. Select the **Remote PasswordOnly** checkbox.
8. Select the **Enabled** checkbox.
9. Click **Apply**. A banner will confirm that the data has been saved.

MODIFY AN EXISTING USER ACCOUNT WITH PASSWORD



1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Edit User** button and make the required changes.
3. Click **Save User**. A banner will confirm the changes have been saved.

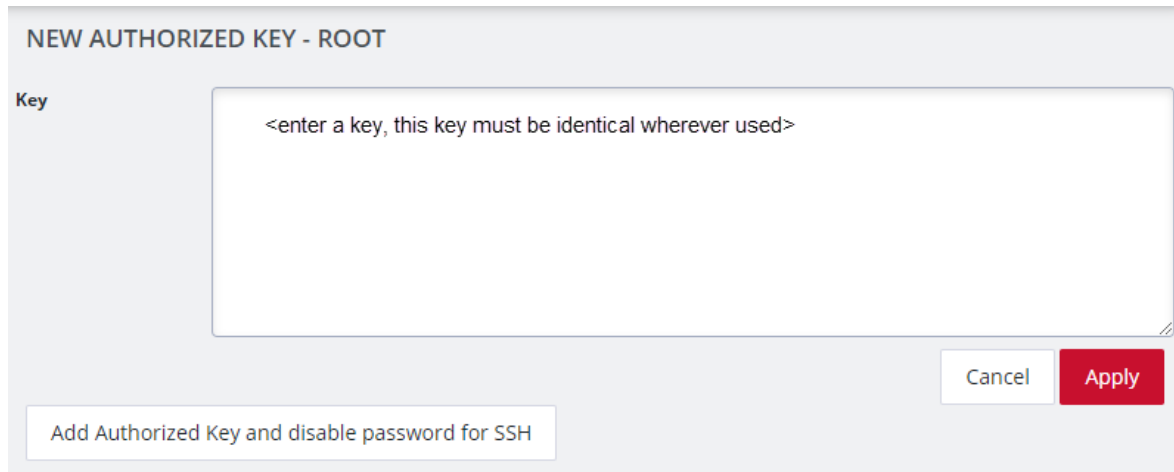
The **Edit Users** dialog allows the user's **Description** to be changed, **Group Memberships** modified, and the user's **Password** to be reset. The username cannot be changed. To disable a user, uncheck the **Enabled** checkbox.

Note:Users of disabled accounts cannot log in to the Console Manager using either the Web-based interface or via shell-based logins.

MANAGE SSH AUTHORIZED KEYS FOR A USER ACCOUNT

To manage SSH authorized keys for a user:

1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Manage SSH Authorized Keys**  button for that user.
3. Click the **Add Authorized Key**  button to add a new key. This opens the **NEW AUTHORIZED KEY** page for this user.



4. Enter the key and click **Apply**. You can also click on **Add Authorized Key** and disable password for SSH for this user from this page.
5. To delete a key, click **CONFIGURE > USER MANAGEMENT > Local Users** and click the **Manage SSH Authorized Key** button for the user.
6. Click the **Delete** button next to the key you wish to remove.

DELETE A USER'S ACCOUNT

To delete a user's account:

1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Delete User** button in the **Actions** section next to the user to be deleted.
3. Click **Yes** in the **Confirmation** dialog.

REMOTE AUTHENTICATION

CONFIGURE > USER MANAGEMENT > Remote Authentication

The Console Manager supports three AAA systems. Select the remote authentication mode to be applied (DownLocal, or Local apply for all modes):

- RADIUS
- TACACS+
- LDAP

Navigate to **CONFIGURE > USER MANAGEMENT > Remote Authentication**, the Remote Authentication Home page is displayed.

REMOTE AUTHENTICATION

Mode
TACACS+

Policy
TACACS+ DownLocal
TACACS+ Local

REMOTE AUTHENTICATION SERVERS

Address Port (defaults to 49)

[+ Add authentication server](#)

TACACS+ login method
PAP

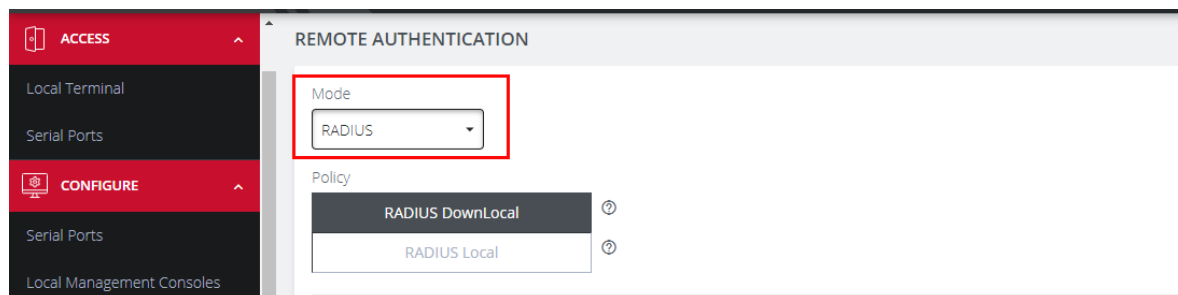
Server password

Confirm server password

Tip: All fields in the Remote Authentication form have tooltips that provide additional information to assist with completing the form fields.

CONFIGURE RADIUS AUTHENTICATION

1. Under **CONFIGURE > User Management > Remote Authentication**, select **RADIUS** from the **Mode** drop-down menu.



The screenshot shows the 'REMOTE AUTHENTICATION' configuration page. The 'Mode' dropdown menu is highlighted with a red box and set to 'RADIUS'. The 'Policy' section shows 'RADIUS DownLocal' and 'RADIUS Local' options. The left sidebar shows the 'CONFIGURE' menu with 'Remote Authentication' selected.

2. Select the preferred Radius Remote Authentication policy to be applied: **Radius DownLocal**, or **Radius Local** (see the tips below).

Tip: RADIUS DownLocal: Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

Tip: RADIUS Local: If remote authentication fails because the user account does not exist on the remote AAA server, the CM attempts to authenticate the user using a local account as per a regular local login

3. Add the **Address** and optionally the **Port** of the authentication server.
4. Add the **Address** and optionally the **Port** of the RADIUS accounting server.
5. Add and confirm the **Server password**, also known as the RADIUS Secret.
6. Click **Apply**.

Note: Multiple servers can be added. The RADIUS subsystem will query them in a round-robin fashion.

To provide group membership, RADIUS needs to be configured to provide a list of group names via the Framed-Filter-Id attribute. The following configuration snippet shows how this can be configured for FreeRADIUS:

```
operator1 Auth-Type := System
```

```
Framed-Filter-ID = ":group_name=west_coast_admin,east_coast_user:"
```

Note: The Framed-Filter-ID attribute must be delimited by the colon character.

CONFIGURE TACACS+ AUTHENTICATION

1. Under **CONFIGURE > USER MANAGEMENT > Remote Authentication**, select TACACS+ from the **Mode** drop-down menu.
2. Select the preferred TACACS+ Remote Authentication policy to be applied: **TACACS+ DownLocal**, or **TACACS+ Local** (see the tips below).

Tip: TACACS+ DownLocal: Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

Tip: TACACS+ Local: If remote authentication fails because the user account does not exist on the remote AAA server, the CM attempts to authenticate the user using a local account as per a regular local login.

3. Add the **Address** and optionally the **Port** of the TACACS+ authentication server to query.
4. Select the **Login Method**. **PAP** is the default method. However, if the server uses DES-encrypted passwords, select **Login**.
5. Add and confirm the **Server password**, also known as the TACACS+ Secret.
6. Add the **Service**. This determines the set of attributes sent back by the TACACS+ server

Note: Multiple servers can be added. The TACACS+ subsystem queries them in a round-robin fashion.

```
user = operator1 {  
    service = raccess {  
        groupname = west_coast_admin,east_cost_user  
    }  
}
```

7. Enable or Disable **Remote Accounting**.

TACACS Accounting is enabled by default, the Remote Auth Server is used as the Accounting server. However one or more Accounting Servers can be specified.

- a. To disable Remote Accounting, select **Disable**
- b. To enable Remote Accounting, select **Enable**.

REMOTE ACCOUNTING

Enable Accounting Disable Accounting

Accounting logs for CLI and Console Port logins will be sent to the first available Remote Authentication Server.

8. Click **Apply**.

Note:For Cisco ACS, see [Setting up permissions with Cisco ACS 5 and TACACS+](#) on the Opengear Help Desk.

CONFIGURE LDAP AUTHENTICATION

1. Under **CONFIGURE > User Management > Remote Authentication**, select **LDAP** from the **Mode** drop-down menu.

2. Select the preferred LDAP Remote Authentication policy to be applied: **LDAP DownLocal**, or **LDAP Local** (see the tips below for explanation).

Tip: LDAP DownLocal: Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

Tip: LDAP Local: If remote authentication fails because the user account does not exist on the remote AAA server, the **CM** will attempt to authenticate the user using a local account as per a regular local login.

2. Add the **Address** and optionally the **Port** of the LDAP server to query.
3. Add the **LDAP Base DN** that corresponds to the LDAP system being queried. For example:

```
CN=example-user,CN=Users,DC=example-domain,DC=com
```

4. Add the **LDAP Bind DN**. This is the distinguished name of a user with privileges on the LDAP system to perform the lookups required for retrieving the username of the users, and a list of the groups they are members of.
5. Input the password for the **LDAP Bind DN** user and confirm the password.
6. Add the **LDAP Username Attribute**. This depends on the underlying LDAP system. Use sAMAccountName for Active Directory systems, and uid for OpenLDAP based systems.
7. Add the **LDAP Group Membership Attribute**. This is only needed for Active Directory and is generally memberOf.

8. If desired, check **Ignore referrals** option. When checked, LDAP will not follow referrals to other remote authentication servers when logging users in. If multiple remote authentication servers exist on the network, checking this option may improve log in times.

Note: Multiple servers can be added. The LDAP subsystem queries them in a round-robin fashion.

LOCAL PASSWORD POLICY

CONFIGURE > USER MANAGEMENT > Local Password Policy

A Password Complexity policy allows network administrators to implement and enforce a password policy that meets the customers' security standards for local users (including root). This functionality enables administrators to mandate the setting of complex passwords thus making it difficult for malicious agents to succeed in password attacks.

Enabling this feature will:

- Enforce the use of complex passwords so as to improve security.
- Schedule expiry of passwords to enforce regular password updates.

Note: Password policy such as complexity and expiry can only be configured by an administrator. Password requirements are applied to all accounts.

Tip: Password policy may be enabled and configured via the Web GUI, rest-api and ogcli. The password policy also applies to underlying CLI tools.

SET PASSWORD COMPLEXITY REQUIREMENTS

[CONFIGURE > USER MANAGEMENT > Local Password Policy](#)

Note: Some password complexity rules are required, other rules are optional. Optional rules can be selected by clicking on the relevant checkbox.

Note: Users are prevented from using the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This password policy applies to the WebUI, Config Shell and CLI. users configured on the system using software versions prior to 23.10 with password “default” are forced to change the user password to something other than “default” after upgrading to 23.10. This password feature update applies to configured boxes with existing users, not just factory defaulted software.

See also "[Password Policy Implementation Rules](#)" on page 183

To set the password complexity requirements:

1. Navigate to [CONFIGURE > USER MANAGEMENT > Local Password Policy](#).
2. Click the **Enforced** button to implement the password complexity policy (the policy is not activated until the **Apply** button is clicked).
3. Enter the information required to form the password complexity rules to comply with your company policy:
 - Password cannot be a palindrome (required)
 - Minimum length (required)

- Must contain an upper case letter (optional)
- Must contain a numeric character (optional)
- Must contain a special character (non-alphanumeric eg. e.g. #,\$,%)
- Disallow user names in passwords (optional)

See "[Password Policy Implementation Rules](#)" on page 183

4. Click the **Apply** button to activate the password complexity policy.

SET PASSWORD EXPIRATION INTERVAL

[CONFIGURE > USER MANAGEMENT > Local Password Policy](#)

See also "[Password Policy Implementation Rules](#)" on the next page

Password Expiration schedules the expiry of passwords to enforce regular password updates. When this feature is applied and a password becomes expired, an expired password prompt is displayed at log-in.

Note: The Password Expiration policy affects local passwords only and does not apply to remote authentication modes.

To set the password expiration interval:

1. Navigate to [CONFIGURE > USER MANAGEMENT > Local Password Policy](#).
2. Click the **Enabled** button to implement the password expiration policy (the policy is not activated until the **Apply** button is clicked).
3. Input a number to represent the desired number of days between mandatory password updates. The default time is 90 days and the minimum is 1 day.
4. Click the **Apply** button to activate the password interval policy.

PASSWORD POLICY IMPLEMENTATION RULES

Rule	Policy
Expiry Rules	<p>The expiry time is measured in number of whole days. When the expiry period is reached users are required to update their password on their next login. The default expiry period is 90 days and the minimum is one (1) day.</p>
	<p>If there are existing user passwords when the expiry is enabled, the expiry time will be applied from when the password was initially set by the user. If a password falls outside the new expiry period the user will be immediately prompted to change the password.</p>
	<p>Local Password policy is only applied to local passwords and does not apply to remote authentication modes.</p>
	<p>When local password policy is enabled it will remain in force until the feature is turned off.</p>
	<p>If the minimum password length is modified and then the password complexity feature is disabled, the minimum length requirement is not updated.</p>
Complexity Rules	<p>The password cannot be a palindrome (this requirement cannot be disabled except by disabling password complexity entirely).</p> <p>(A palindrome is a word or other sequence of characters that reads the same backward as forward, such as <i>madam</i> or <i>racecar</i>).</p>
	<p>The minimum length (enforced) must be at least 8 char-</p>

	acters (this requirement cannot be disabled except by disabling password complexity entirely).
	The password should contain at least one upper case alphabetic character (enabled or disabled separately).
	The password must contain at least one numeric character (enabled/disabled separately).
	The password should contain at least one special character (e.g. #,\$,%) (enabled/disabled separately).
	The password cannot contain your user-name.
	Complexity requirements will apply when a user next tries to update their password.
	An administrator can force the expiry of a users password by running the ogCLI command: <code>passwd --expire {username}</code> to force a user to change their password.
	The operations <code>ogadduser</code> , <code>ogpasswd</code> and <code>ogsshaddsshkey</code> have been removed. You should instead use ogCLI for these operations.

SERVICES

CONFIGURE > SERVICES

23.10.3	CONFIGURE Menu	184
---------	----------------	-----



The **CONFIGURE > SERVICES** menu lets you manage services that work with the Console Manager.

23.10.3	CONFIGURE Menu	185
---------	----------------	-----

FIPS COMPLIANCE

The Federal Information Processing Standard Publication 140-2 (FIPS 140-2) is a U.S. government computer security standard that is used to approve cryptographic modules. Opengear appliances operating in FIPS mode provide FIPS 140-2 level one compliance by utilizing FIPS validated OpenSSL 3.0.8 cryptographic library while in FIPS mode.

Note: The default provider will be 3.0.10, however, the FIPS provider remains on 3.0.8 in release 23.10.3. See the example of list providers later in this topic under the section ["Verify that FIPS is enabled" on the next page.](#)

CONFIGURE FIPS

Enable FIPS mode at the CLI as follows:

ENABLE FIPS

Enable FIPS via config shell:

```
root@<device name>:~# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
config: system/fips
config(system/fips): enabled true
config(system/fips): apply
Updating entity system/fips.
```

Enable FIPS via ogcli:

23.10.3	FIPS Compliance	186
---------	-----------------	-----

```
ogcli update system/fips enabled=true
```

DISABLE FIPS

Disable FIPS via config shell:

```
root@<device name>:~# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
config: system/fips
config(system/fips): enabled false
config(system/fips): apply
Updating entity system/fips.
```

Disable FIPS via ogcli:

```
ogcli update system/fips enabled=false
```

VERIFY THAT FIPS IS ENABLED

1. Check the OpenSSL FIPS providers.

```
root@<device name>:~# openssl list -providers
Providers:
default
  name: OpenSSL Default Provider
  version: 3.0.10
  status: active
fips
```

```
name: OpenSSL FIPS Provider
version: 3.0.8
status: active
```

2. Check that the digest algorithms provided by OpenSSL is limited to FIPS compliant ciphers/algorithms.

```
root@<device name>:~# openssl list -digest-algorithms
...
Provided:
  { 2.16.840.1.101.3.4.2.1, SHA-256, SHA2-256, SHA256 } @ default
  { 2.16.840.1.101.3.4.2.10, SHA3-512 } @ default
  { 2.16.840.1.101.3.4.2.8, SHA3-256 } @ default
  { 2.16.840.1.101.3.4.2.7, SHA3-224 } @ default
  { 2.16.840.1.101.3.4.2.2, SHA-384, SHA2-384, SHA384 } @ default
  { 2.16.840.1.101.3.4.2.3, SHA-512, SHA2-512, SHA512 } @ default
  { 2.16.840.1.101.3.4.2.5, SHA-512/224, SHA2-512/224, SHA512-224 } @
default
  { 2.16.840.1.101.3.4.2.12, SHAKE-256, SHAKE256 } @ default
  { 1.3.14.3.2.26, SHA-1, SHA1, SSL3-SHA1 } @ default
  { 2.16.840.1.101.3.4.2.9, SHA3-384 } @ default
  { 2.16.840.1.101.3.4.2.11, SHAKE-128, SHAKE128 } @ default
  { 2.16.840.1.101.3.4.2.4, SHA-224, SHA2-224, SHA224 } @ default
  { 2.16.840.1.101.3.4.2.6, SHA-512/256, SHA2-512/256, SHA512-256 } @
default
  { KECCAK-KMAC-128, KECCAK-KMAC128 } @ default
  { KECCAK-KMAC-256, KECCAK-KMAC256 } @ default
  { 2.16.840.1.101.3.4.2.1, SHA-256, SHA2-256, SHA256 } @ fips
  { 2.16.840.1.101.3.4.2.10, SHA3-512 } @ fips
  { 2.16.840.1.101.3.4.2.8, SHA3-256 } @ fips
```

```
{ 2.16.840.1.101.3.4.2.7, SHA3-224 } @ fips
{ 2.16.840.1.101.3.4.2.2, SHA-384, SHA2-384, SHA384 } @ fips
{ 2.16.840.1.101.3.4.2.3, SHA-512, SHA2-512, SHA512 } @ fips
{ 2.16.840.1.101.3.4.2.5, SHA-512/224, SHA2-512/224, SHA512-224 } @
fips
{ 2.16.840.1.101.3.4.2.12, SHAKE-256, SHAKE256 } @ fips
{ 1.3.14.3.2.26, SHA-1, SHA1, SSL3-SHA1 } @ fips
{ 2.16.840.1.101.3.4.2.9, SHA3-384 } @ fips
{ 2.16.840.1.101.3.4.2.11, SHAKE-128, SHAKE128 } @ fips
{ 2.16.840.1.101.3.4.2.4, SHA-224, SHA2-224, SHA224 } @ fips
{ 2.16.840.1.101.3.4.2.6, SHA-512/256, SHA2-512/256, SHA512-256 } @
fips
{ KECCAK-KMAC-128, KECCAK-KMAC128 } @ fips
{ KECCAK-KMAC-256, KECCAK-KMAC256 } @ fips
```

CONSIDERATIONS FOR USING THE FIPS FEATURE

In organizations where FIPS is required, the following points should be noted:

- OpenSSL 3.0.8 FIPS provider limits the available cryptography ciphers/algorithms only those that have been validated by laboratory to be FIPS compliant.

Caution: Configuration backup should be taken before enabling or disabling FIPS.

Caution: FIPS has the potential to break any service with secure connectivity, including services listed in the following table:

23.10.3	FIPS Compliance	189
---------	-----------------	-----

Feature	Affected Process/Service	Impact
Lighthouse enrollment	OpenVPN	OpenVPN is not compliant with FIPS standards; this issue is a recognized problem specifically when OpenSSL 3.x is being used. Once OpenVPN addresses this issue, it will also meet FIPS compliance standards. However, for compatibility with Lighthouse enrollment, this feature remains enabled despite the non-compliance.
IPsec	Strongswan	Needs to be operated in FIPS mode to be FIPS compliant. The other end of the tunnel does not need to be operating FIPS mode to connect.
Remote authentication	freeradius, tacacs, ldap	These are not FIPS compliant.
NTP	chrony	Authenticated NTP servers with MD5 will not connect. Use an algorithm that is FIPS compliant.

SNMP	ogtrapd, snmpd, snmptrapd	Authentication and Encryption should be used as the security policy as V1 and V2 have no encryption. SNMPv3 with MD5 encryption will fail. Use an algorithm that is FIPS compliant. It is recommended that authPriv security policy is used when in FIPS mode for SNMPv3.
LDAP	OpenSSL	LDAP has no encryption, therefore it does not use OpenSSL. For FIPS compliance it is recommended that it is not used.
OpenSSL	OpenSSL MD5	When OpenSSL MD5 is not available, pam_tacplus uses its own implementation of MD5. When FIPS is enabled it does not use OpenSSL (but will continue to work). Therefore, it is recommended that it is not used in FIPS mode.
SMF	SMF	Use of the SMF feature will render the device non-com-

		pliant for FIPS.
SSH connections	SSH	For SSH connections, a FIPS compliant algorithm must be specified as part of the command to connect. See the note below:
NetOps Modules	gre (Secure Provisioning) nom-ipaccess-lhvpn (IP access) nom-ag-lhvpn (Access Gateway)	Opengear NetOps Modules are not functional when FIPS mode is enabled.
<p>Note: SSH will require the cipher to be manually specified when FIPS is enabled. e.g. ssh root@10.0.0.1 -c aes256-gcm@openssh.com</p>		
Wireguard		Wireguard is not FIPS compliant and should not be used in FIPS mode.
Routing protocols		Routing protocols should not select an MD5 cipher.

BRUTE FORCE PROTECTION

[CONFIGURE > SERVICES > Brute Force Protection](#)

23.10.3	FIPS Compliance	192
---------	-----------------	-----



A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until the one correct combination that works.

Brute Force Protection offers an essential defense mechanism by automatically blocking access from offending source IP addresses.

Caution: Brute Force Protection may prevent access to the system during an emergency.

CONFIGURE BRUTE FORCE PROTECTION

Note: Brute Force Protection is enabled by default for SSH and Web UI.

To configure Brute Force Protection:

1. Navigate to **CONFIGURE > SERVICES > Brute Force Protection**.
2. Choose the desired settings as described below.
3. Click **Apply** to save the changes.

Field	Values	Description
SSH Protection	Enabled / Disabled	Enable Brute Force Protection for SSH login attempts.
HTTPS Protection	Enabled / Disabled	Enable Brute Force Protection for Web UI login

Field	Values	Description
		attempts.
Maximum failed attempts	Attempts: 3 (minimum) Time period in minutes: 1 (minimum)	The number of failed access attempts permitted within the given time period before preventing access.
Lockout period	60 (minimum)	The number of seconds that an IP address will be banned after violating the Brute Force Protection policies.

VIEWING CURRENT BANS

IP addresses that are currently blocked appear in the CURRENT BANS section of the Web UI, displaying the address and remaining duration of the ban or how long ago the ban was lifted.

Hover over the ban time for more detailed information.

23.10.3	FIPS Compliance	194
---------	-----------------	-----

CURRENT BANS

10.0.0.150

The ban was removed [a minute ago](#)

10.0.0.151

The ban was removed [a minute ago](#)

10.0.0.152

The ban was removed

Banned since:
Tue Sep 14 2021 16:15:50 GMT-0600

10.0.0.153

The ban was removed [a few seconds ago](#)

MANAGING BRUTE FORCE PROTECTION VIA COMMAND LINE

For more control over Brute Force Protection, administrative users can use the command line to configure the service and remove bans manually.

Description	Command	Notes
Display Brute Force Protection configuration	<pre>ogcli get services/brute_force_protection</pre>	
Update Brute Force Protection configuration	<pre>ogcli replace services/brute_force_protection << END ban_time=180</pre>	Ban time in seconds. Find time in minutes.

Description	Command	Notes
	<pre> find_time=1 https_enabled=false max_retry=4 ssh_enabled=true END </pre>	
Un-ban an IP address	<pre> fail2ban-client unban <ipaddress> </pre>	
Un-ban all current bans	<pre> fail2ban-client unban --all </pre>	
List SSH bans	<pre> fail2ban-client status sshd </pre>	SSH protection must be enabled
List HTTPs bans	<pre> fail2ban-client status https </pre>	HTTPs protection must be enabled
List all bans with ogcli	<pre> ogcli get monitor/brute_ force_protection/bans </pre>	

HTTPS CERTIFICATE

[CONFIGURE](#) > [SERVICES](#) > [HTTPS Certificate](#)

The Console Manager ships with a private SSL Certificate that encrypts communications between it and the browser.

To examine this certificate or generate a new Certificate Signing Request, select **CONFIGURE > SERVICES > HTTPS Certificate**. The details of the **Current SSL Certificate** are shown on the landing page.

CURRENT SSL CERTIFICATE

Common Name ⓘ

default

The group overseeing this device.

Tool tips assist with completing the form

Organizational Unit ⓘ

Organization ⓘ

Locality/City ⓘ

State/Province ⓘ

Country ⓘ

US

Email ⓘ

Key Length (bits) ⓘ

2048

Issue Date ⓘ

Apr 26 20:11:11 2021 GMT

Expiry Date ⓘ

Apr 27 20:11:11 2022 GMT

Below this listing is a **Certificate Signing Request** form, which can be used to generate a new SSL certificate. Complete the form, then click **Apply**.

23.10.3	FIPS Compliance	197
---------	-----------------	-----

CERTIFICATE SIGNING REQUEST

Common Name ?

The group overseeing this device.

Tool tips assist with completing the form content

Organizational Unit ?

Organization ?

Locality/City ?

State/Province ?

Country ?

Email ?

Key Length (bits) ?

Challenge Password ?

Confirm Password ?

Private Key File ?

 No file chosen

Apply

NETWORK DISCOVERY PROTOCOLS

[CONFIGURE](#) > [SERVICES](#) > Network Discovery Protocols

23.10.3	FIPS Compliance	198
---------	-----------------	-----



The Console Manager displays LLDP/CDP Neighbors when enabled for a connection. See **CONFIGURE > SERVICES > Network Discovery Protocols** to enable/disable.

The **CONFIGURE > SERVICES > Network Discovery Protocols > LLDP/CDP NEIGHBORS** page allows you to enable this service by clicking the **Enabled** checkbox.

You can set a System Description that overrides the default system description sent by the network discovery protocol daemon. The default description is the kernel name, the node name, the kernel version, the build date and the architecture.

A value can be entered in the CDP Platform Override to override the CDP platform name. The default name is the kernel name (Linux).

NETWORK DISCOVERY PROTOCOLS

SETTINGS

Enabled

Link Layer Discovery Protocol (LLDP) and Cisco Discovery Protocol (CDP).

System Description Override

Use this setting to override the default system description

This setting overrides the default system description sent by the network discovery protocol daemon. The default description is the kernel name, the node name, the kernel version, the build date and the architecture.

CDP Platform Override

Use this setting to override the default CDP platform name

This setting overrides the CDP platform name. The default name is the kernel name (Linux).

Select one or more checkboxes in the **NETWORK INTERFACES** section of the page and click **Apply**.

23.10.3	FIPS Compliance	199
---------	-----------------	-----

NETWORK INTERFACES

Selecting an interface allows LLDP/CDP monitoring for that interface.

NET1 - 1G Copper/SFP

NET2 - 1G Copper/SFP

Apply

FILE SERVER

[CONFIGURE](#) > [SERVICES](#) > [File Server](#)

The Console Manager can be configured to serve files to clients via Trivial File Transfer Protocol (TFTP).

TFTP can be used by nodes on the network to perform a network boot, or to allow backup and restore of configuration files.

Note: Limitations

- The user is responsible for disk space management.
- User permissions cannot be set on files at this time.

23.10.3	FIPS Compliance	200
---------	-----------------	-----

ENABLE TFTP SERVICE

Note:The TFTP service is disabled by default.

To enable the TFTP service:

- Click the **TFTP Enabled** button.



- Click **Apply** to save the changes.
- The TFTP service is now running with a default location of **/mnt/nvram/srv**.
This location is where all files uploaded to the TFTP server will be stored.

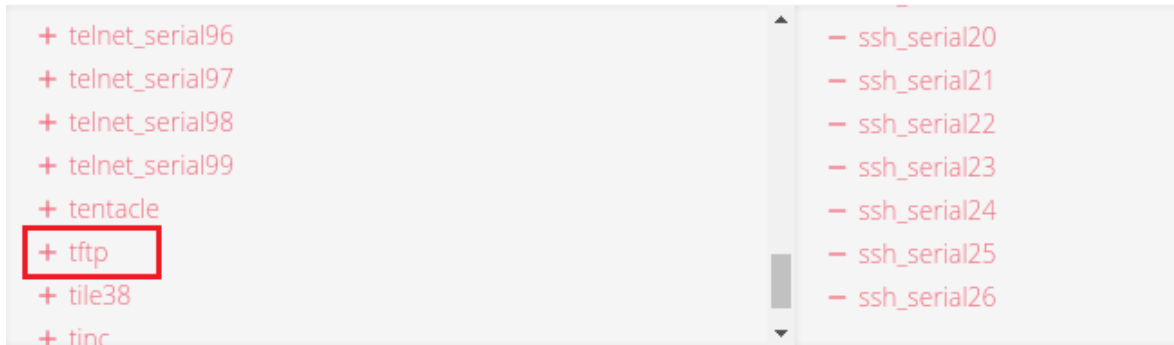
Note:The disk space usage information displayed on the page indicates the usage of the whole storage volume.

MODIFY FIREWALL ZONES TO ALLOW THE TFTP SERVICE TO BE USED

The TFTP service must be allowed through a firewall zone so that clients may upload and retrieve files.

- Navigate to the Firewall Management page via **CONFIGURE > FIREWALL > Management**.
- Expand the desired firewall zone and click the **Edit Zone** button.
- Allow the "tftp" service from the list of Permitted Services.

Permitted Services



- Click **Apply** to save the changes.
- On the File Server page, the zones with TFTP enabled are now displayed.

ZONES WITH TFTP ENABLED

LAN , WAN

UPDATE THE TFTP SERVICE STORAGE LOCATION

The location used by the TFTP service can be updated using the **ogcli** tool.

Note: The storage location must be an existing directory before running ogcli update.

Caution: Using a storage volume other than **/mnt/nvram** is not recommended. Data may be lost after reboot, or be inaccessible when switching boot slots.

- As an administrative user, run:

```
ogcli update services/tftp path=\"<new path>\"
```

ROUTING

[CONFIGURE > SERVICES > Routing](#)

The Console Manager supports Static Routing and Dynamic Routing. Static Routing is currently configured via the ogcli interface, while Dynamic Routing is configured via the UI.

DYNAMIC ROUTING

To enable Dynamic Routing on the CM, navigate to the **CONFIGURE > SERVICES > Routing** page.

Dynamic Routing supports four routing protocols, these are:

- BGP (Border Gateway Protocol)
- OSPF (Open Shortest Path First Protocol) (see "[Routing](#)" above later in this topic).
- IS-IS (Intermediate System to System Protocol)
- RIPD (Routing Information Protocol)

Select the preferred routing protocol then click **Apply**.

Note: If no protocol is selected, no route sharing services are run on the CM.

ROUTING

DYNAMIC ROUTING PROTOCOL

- BGP (Border Gateway Protocol)
- OSPF (Open Shortest Path First Protocol)
- IS-IS (Intermediate System to System Protocol)
- RIPD (Routing Information Protocol)

Apply

STATIC ROUTING (VIA THE OGCLI)

To enable Static Routing on the CM, open an ogcli terminal by navigating to **ACCESS > Local Terminal**.

STATIC ROUTING OGCLI HELP

For Help on implementing a Static Route protocol via ogcli, enter the command:

```
ogcli help static_routes
```

CREATE STATIC ROUTE - EXAMPLE:

```
ogcli create static_route << 'END'  
destination_address="10.1.45.0"  
destination_netmask=24  
gateway_address="192.168.1.1"  
interface="system_net_physifs-1"  
metric=100  
END
```

STATIC ROUTING ARGUMENTS

Argument	Description
<code>get</code>	Get a list of static routes.
<code>create</code>	Add a static route.
<code>replace</code>	Similar to the "Create Static Route" example given on the previous page. Creates a single static route by specifying its UUID; or a list of static routes. Overwrites existing routes.
<code>delete</code>	Delete all static routes.
<code>merge</code>	Merge the existing configuration list with a new list.

OSPF CONFIGURATION

Open Shortest Path First (OSPF) is a link-state routing protocol used to discover routes on a network. It is used to dynamically adjust routes on the Console Server so that subnets connected to different interfaces can reach each other by routing through the Console Server.

Support for OSPF configuration and WireGuard was added to the REST API and Config Shell at release 23.02.

Caution: Users are discouraged from editing OSPF configuration when it has been marked as managed by a Lighthouse. A warning message is displayed when an attempt is made to edit any configuration pushed down from Lighthouse through config shell. After being warned of the risk users may continue to edit configuration with a **managed_by** field set through config shell.

 This zone is managed by **Lighthouse** and cannot be edited.

MANAGED CONFIGURATION ITEMS

Certain items in the configuration can contain an optional **managed_by** field. Configuration items that have the `managed_by` field set are considered to be "managed". The `managed_by` field is set by a managing entity such as lighthouse, when the network plan is being managed by a remote node.

The following features can have managed configuration:

- Firewall Zones
- Firewall Policies
- Routing OSPF
- WireGuard Tunnels

If a firewall zone, policy or WireGuard tunnel is managed, this does not affect sister contexts, for example, if the WireGuard tunnel is managed, any other WireGuard tunnels configured separately by the user are not managed. However, there is only one OSPF configuration file and users will need to bypass the **managed_by** field in config shell in order to edit the configuration.

NEW FIELDS IN REST API & CONFIG SHELL

REST API

The OSPF sub-object now has a number of new fields:

```
"services": {
  "routing": {
    "bgpd": {
      "enabled": true
    },
    "isisd": {
      "enabled": false
    },
    "ripd": {
      "enabled": true
    },
    "ospfd": {
      "enabled": false,
      "router_id": "",
      "redistribute_connected": false,
      "redistribute_static": false,
      "redistribute_kernel": false,
      "interfaces": [],
      "neighbors": [],
      "networks": []
    }
  }
}
```

CONFIG SHELL

The services/routing OSPF context has new fields similar to the REST API:

```
config(services/routing ospfd): show
Entity services/routing field ospfd
  enabled                               false
  redistribute_connected false
  redistribute_static                   false
  router_id                             ""
  interfaces (array)
  neighbors (array)
  networks (array)
```

Field	Condition	Definition
enabled	(true / false)	When set to true, the OSPF service is started.
redistribute_connected	(true / false)	If this option is enabled, any directly connected network routes will be broadcast to OSPF neighbours
redistribute_static	(true / false)	Network routes can be statically defined (in OSPF, not the Linux Kernel) by editing the ospfd.conf file or through <code>vttysh</code> . If this option is enabled, redistribute_routes broadcasts any static routes that are managed by OSPF.

redistribute_kernel	(true / false)	If this option is enabled, network routes that are configured in the Linux kernel via DHCP or static definition will be shared with OSPF neighbors.
router_id		The router id (RID) is a 32-bit number which must be expressed as a dotted quad (i.e. in the format A.B.C.D). The RID is used to identify the router. It must be unique within the OSPF network. The highest RID in the network will be used to determine which OSPF node is the designated router.

INTERFACES, NEIGHBORS AND NETWORKS.

There are a number of sub-objects under the ospfd context: interfaces, neighbors and networks.

INTERFACES CONTEXT

The services/routing OSPF interfaces context is an array in which each element holds the specific individual interface related parameters for OSPF. Each interface has the following fields:

```
Entity services/routing field ospfd interfaces 0
  auth_method      ""      (required)
  cost              ""
  priority         ""
  name             ""      (required)
  non_broadcast    ""      (required)
  passive          ""      (required)
```

Definitions of interface related parameters for OSPF:

Parameter	Definition
auth_method	<p>The authentication method to use for communications on this interface. Should be one of 'no_auth', 'cleartext' or 'md5'. If authentication is enabled (i.e. not no_auth), one or multiple authentication keys can be configured depending on your authentication method chosen.</p> <ul style="list-style-type: none"> • Cleartext authentication only needs one authentication key. • Md5 authentication can use multiple authentication keys, each of which needs a unique id.
cost	<p>The link cost of the interface used in OSPF route calculations. It is normally auto-calculated, but can be specified manually in the range of 1 to 65535.</p>
priority	<p>The priority of a router on an OSPF interface mainly is used to determine the designated router/backup designated router (DR/BDR) for a network. OSPF forwards all messages to the designated router, reducing the amount of repetitive routing traffic on the network. The priority is in the range of 0 to 255. The default priority for each router is 1 unless specified. Selecting a priority of 0 makes the router unable to become a DR/BDR. The higher the priority, the higher chance a OSPF router has of winning the DR/BDR election.</p>
name	<p>The name of the interface these settings apply to. This should match the name of an interface on the device.</p>
non_broadcast	<p>May be true or false. If true, the interface will be marked as</p>

	non broadcast for OSPF purposes. This would mean OSPF would not use multicast on this interface, and static neighbours would need to be defined.
passive	May be true or false. If true, the interface should be marked as passive for OSPF purposes. This would mean LSAs are not traded on this link.

NEIGHBORS CONTEXT

The services/routing OSPF neighbors context is an array where each element holds details about adjacent static neighbor devices. Neighbors must be specified for non-broadcast networks.

```
config(services/routing ospfd neighbors): add
config(services/routing ospfd neighbors 0): show
Entity services/routing field ospfd neighbors 0
address "" (required)
```

Where `address` is an IPv4 host address of the static neighbor.

NETWORKS CONTEXT

The services/routing OSPF networks context is an array where each element holds IP network configurations to enable the system OSPF service for:

```
config(services/routing ospfd networks): add
config(services/routing ospfd networks 0): show
Entity services/routing field ospfd networks 0
address_with_mask "" (required)
area "" (required)
```

Network Configuration	Definition
address_with_mask	An IPv4 network address with CIDR subnet mask to enable OSPF for (e.g. A.B.C.D/E). No host bits should be set.
area	An OSPF network can be divided into sub-domains or groupings called areas which limit the scope of route information distribution. We specify the area number/id we want the interface to be in. This can be an integer between 0.0.0.0 and 255.255.255.255 or can take a form similar to an IP address A.B.C.D. All routers inside an area must be a part of the same OSPF network and have the same area number/id to become OSPF neighbours.

INTERACTION WITH CONFIGURATION FILES

The first line of `/etc/quagga/ospfd.conf` controls whether the console server configuration system will overwrite the file with new content or keep custom user configuration. This supports customers who want to upload a custom configuration file for OSPF. If the first line contains only the text `! autogen`, the configuration system will overwrite the file, otherwise, the configuration system will have no effect.

To verify the OSPF configuration, the configuration file generated can be found in `/etc/quagga/ospfd.conf`:

```
! autogen
! This configuration file has been autogenerated. Any changes made
within
! will be overwritten. To stop this and allow for manual editing,
remove
! or change the first line of this file to something other than '!
```

```
autogen'.  
  
! The behaviour can be reenabled by restoring the first line to this  
or by  
  
! completely removing this contents of this file.  
  
!  
interface wg-smf-1  
ip ospf network non-broadcast  
  
!  
interface net1  
  
!  
router ospf  
ospf router-id 0.0.0.1  
log-adjacency-changes  
redistribute connected  
redistribute static  
network 10.0.0.0/24 area 0.0.0.0  
network 192.168.41.0/24 area 0.0.0.0  
neighbor 10.0.0.1  
  
!  
line vty  
  
!
```

CONFIRM OSPF NEIGHBOURS

Use the `vttysh` command line tool to see if OSPF neighbours have been discovered:

23.10.3	FIPS Compliance	213
---------	-----------------	-----

```
root@om2200-q:~# vtysh -c 'show ip ospf neighbor'
Neighbor ID          Pri State Dead Time Address          Interface
RXmtL              RqstL              DBsmL
- 0 Attempt/DROther          33.007s 10.0.0.1 wg-smf-1:10.0.0.2
0 0 0
```

(Where `wg-smf-1` is a user-named interface).

WIREGUARD CONFIGURATION

WireGuard is an open source encrypted VPN solution; WireGuard configuration support was added to the REST API and Config Shell at release 23.8. WireGuard facilitates communication between two peer devices; in order to communicate with a peer, both devices must have a virtual WireGuard interface configured over the physical or virtual interface they are connected over.

Note:Users who have pre-existing configuration files for WireGuard will not have their configurations overwritten as the configurator will only modify those files if they are initially missing or are prefixed with a disclaimer that manual edits will be overwritten.

VIEWING A WIREGUARD CONFIGURATION

WireGuard installs the **wg** tool which can be used to control, configure and monitor WireGuard . Refer to the WireGuard online tools index page: [index : wireguard-tools](#)

Note:OpenGear does not own or operate the WireGuard tools web page and is not responsible for its content or maintenance. The link is provided only for the reader's convenience.

CONFIGURE WIREGUARD THROUGH CONFIG SHELL OR REST API

WireGuard is configured through Config Shell or REST API. The minimum configuration of WireGuard is shown in the following:

1. Provide a name for the interface (wg0 in the example below).
2. Set enabled.
3. Set the `private_key` of your WireGuard interface.
4. Add an address (at least one) for your WireGuard interface (10.0.0.1/24 in this case).
5. Add a peer with the following parameters: `endpoint_address`, `endpoint_port`, `public_key`.
6. Add an `allowed_ip` for your peer. At least one - this is the WireGuard address(es) (as it can also accept an address range) of the other interface to which you are connected.

For example:

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): private_key
AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
config(wireguard wg0): enabled true
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
config(wireguard wg0 addresses): up
config(wireguard wg0): peers
config(wireguard wg0 peers): add
```



```
config(wireguard wg0 peers 0): public_key
o+quB4sbUAG2hEGSPpMNTn00YSaQTP7dD+Q4IVjicW8=
config(wireguard wg0 peers 0): allowed_ips
config(wireguard wg0 peers 0 allowed_ips): add 10.0.0.2/32
config(wireguard wg0 peers 0 allowed_ips): up
config(wireguard wg0 peers 0): endpoint_address 192.168.1.2
config(wireguard wg0 peers 0): endpoint_port 51820
config(wireguard wg0 peers 0): up
config(wireguard wg0 peers): top
```

CONFIG SHELL WIREGUARD CONFIGURATION

The following shows a typical WireGuard configuration in Config Shell:

```
config: show wireguard wg0
Entity wireguard item wg0
  description ""
  enabled true
  mtu 1420
  name wg0
  port 51820
  private_key AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
  public_key ""
  table ""
  addresses (array)
    0 10.0.0.1/24
  peers (array)
    0 (object)
      endpoint_address 192.168.1.2
```

```
endpoint_port 51820
keep_alive ""
public_key o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=
allowed_ips (array)
    0 10.0.0.2/32
post_down_hooks (array)
post_up_hooks (array)
pre_down_hooks (array)
pre_up_hooks (array)
```

REST API WIREGUARD CONFIGURATION

The following shows a typical WireGuard configuration in Config Shell:

```
{
  "wireguards": [
    {
      "enabled": true,
      "post_down_hooks": [],
      "id": "wireguard_tunnels-1",
      "pre_up_hooks": [],
      "post_up_hooks": [],
      "private_key":
"AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=",
      "name": "wg0",
      "pre_down_hooks": [],
      "addresses": [
        "10.0.0.1/24"
      ],
    },
  ],
}
```

```
"peers": [  
  {  
    "allowed_ips": [  
      "10.0.0.2/32"  
    ],  
    "public_key":  
"o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=",  
    "endpoint_address": "192.168.1.2",  
    "endpoint_port": 51820  
  }  
]  
}  
]
```

CONFIGURABLE WIREGUARD FIELDS

The WireGuard <interface-name> context holds the configuration for a WireGuard connection. The following fields can be configured:

WireGuard Field	Description
description	This can be any user text to describe the WireGuard interface.
enabled	Values may be true or false . When enabled, WireGuard will be started for this configuration.

mtu	Allows customization of the maximum transmission unit (MTU) for the local WireGuard interface. The range is 1280 - 1472 and if not set, WireGuard will use the internal default of 1420.
name	The name of the WireGuard interface used in the Linux kernel. Names must be unique, max 15 characters and only contain letters, numbers, hyphens or underscores.
port	The port the local instance of WireGuard will listen on. The range is 1 to 65535 and defaults to 51820.
private_key	The private key to use to authenticate the local WireGuard interface. This is obtained by running the wg genkey command.
public_key	The public key that corresponds your private key, which WireGuard peers will authenticate with. This is obtained by running the wg pubkey command.
table	The routing table for the WireGuard routes. Can be a table number, 'off' or 'auto'.

WIREGUARD CONTEXT SUB-OBJECTS

There are a number of sub-objects under the WireGuard context: addresses, peers and hooks.

ADDRESSES

The wireguard <interface-name> addresses context is a list that holds the IPv4 CIDR addresses of the local Wireguard interface. These are statically assigned when the WireGuard interface is brought up.

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
```

PEERS

The following list defines the WireGuard settings for Wireguard-capable remote peers. Each peer has the following fields:

```
config(wireguard wg0 peers 0): show
Entity wireguard item wg0 field peers 0
  endpoint_address ""
  endpoint_port ""
  keep_alive ""
  public_key "" (required)
  allowed_ips (array) (required)
```

Peer Field	Description
endpoint_address	A reachable IP address or fully-qualified domain name for the remote peer with a WireGuard interface.
endpoint_port	The port number for which the WireGuard instance is listening on the remote peer.
keep_alive	Equivalent to PersistentKeepalive in the WireGuard config, this specifies how often the WireGuard interface must send a keep alive packet. This helps keep the routing entry alive for scenarios where the peer is behind a NAT.

public_key	The public key that will be accepted by the local WireGuard service if offered by a peer for the purpose of mutual authentication during a five step key exchange process.
allowed_ips -	A list which specifies the IP ranges for which a peer routes traffic. For multiple WireGuard interfaces on the same device, the addresses must not overlap. The IP addresses specified here are the addresses of the peer's WireGuard interface(s) - this is where the peer "routes traffic". These are specified as IPv4 addresses in a.b.c.d/<cidr_mask> format.

HOOKS

WireGuard allows for commands to be executed before/after the interface is brought up/down. These can be specified in the following array fields:

Note: Each field is an array of strings that correspond to commands to be executed.

Hook	Description
pre_up_hooks	Run a command before the interface is brought up (optional).
post_up_hooks	Run a command after the interface is brought up (optional).
pre_down_hooks	Run a command before the interface is brought down (optional).

post_down_hooks

Run a command after the interface is brought down (optional).

ADDING A WIREGUARD INTERFACE TO A FIREWALL ZONE

The WireGuard interface can be added to a firewall zone as in the following example:

```
Entity firewall/zone item zone
description "" (required)
label "" (required)
masquerade "" (required)
name zone
permit_all_traffic "" (required)
address_filters (array)
custom_rules (array)
physifs (array)
port_forwarding_rules (array)
wireguards (array)
```

SSH

CONFIGURE > SERVICES > SSH

To modify the properties of the port used for connecting to serial consoles via SSH, navigate to **CONFIGURE > SERVICES > SSH** .

The following table gives the definitions of the configurable SSH properties.

23.10.3	FIPS Compliance	222
---------	-----------------	-----

Parameter	Definition
Serial Port Delimiter	The delimiting character used to separate the username with port selection information. The default delimiter is a plus sign (+). For example, username+port@address.
Port Number for Direct SSH Links	If SSH is configured to be reachable on a non-standard port, the Direct SSH links on the serial ports page will use this port number.
Max Startups Start	The number of unauthenticated connections before they are refused.
Max Startups Rate	This is the percentage of unauthenticated connections refused. This percentage is a probability that increases linearly until the unauthenticated connections reach full.
Max Startups Full	The number of unauthenticated connections allowed.

UNAUTHENTICATED ACCESS TO SERIAL PORTS

For information about Unauthenticated Access to Serial Ports, see "[Unauthenticated SSH to Serial Ports](#)" below.

UNAUTHENTICATED SSH TO SERIAL PORTS

[Configure](#) > [Services](#) > [SSH](#)

The Unauthenticated SSH Access feature provides the option to access console ports (using TCP high ports) by establishing per-port SSH connection between a



console and serial ports at a remote device. This allows a single step log-in and avoids the necessity for two log-ins to reach a remote end device within secure, closed networks.

Usually, you would need to authenticate on the Opengear appliance, followed by any log in to a device you are connecting to via the serial port.

When unauthenticated access is enabled SSH is available to all serial ports on the device without requiring a password.

Note: Unauthenticated access can be used with or without IP aliases for serial ports.

Caution: For security, **Unauthenticated SSH** should only be used when operating within a trusted, closed network, for example within a lab. There is a security risk in allowing any kind of unauthenticated access to serial ports and any terminals connected to them.

ENABLE UNAUTHENTICATED SSH

Authenticated or Unauthenticated access is determined via a global configuration option. Unauthenticated access to individual ports is achieved by command such as `ssh -p 300X user@<IP>`.

ENABLE SSH

Note: This feature may be enabled using the default settings without the need for configuration.

1. Open the SSH form, **Configure > Services > SSH > SSH (form)**.
2. Complete the SSH form (if this is the first time Unauthenticated SSH has been used), a description of the input data is provided at "[Properties and Settings](#)" on page 227 in this topic.
3. When required, enable the Unauthenticated SSH feature by clicking the **Enabled** button.

Note: Unauthenticated access to all serial ports will be available through SSH on TCP port 3000+ or Serial Port IP aliases.

ENABLE/DISABLE

Enabling or disabling this feature is done in the user interface.

To **enable** the feature click on the **Enabled** button then click the **Apply** button. The feature is enabled immediately and a pop-up will confirm that the feature is enabled.

Note: Clicking the **Apply** button saves any changes you have made to the SSH form. A Details Saved banner confirms that the changes have been saved.

To **disable** the feature click on the **Disabled** button then click the **Apply** button. There is no confirmation pop-up when the feature is disabled.

CONNECTING DIRECTLY TO SERIAL PORTS

For ports that have been configured with the SSH access service, you can connect directly to a port and start a session, bypassing the chooser, by using one of the conventions described in the following:

Convention	Example
<p>Use a network client to connect to the service network Base Port + serial port number.</p>	<pre data-bbox="675 688 1378 842"># SSH to serial port 1 by TCP port ssh -p 3001 -l operator 70.33.235.190</pre> <p data-bbox="675 877 1378 947">In this example, the SSH base port is TCP port 3000, so SSH to TCP port 3001 directly connects you to serial port 1</p>
<p>SSH to the Opengear node, log in adding +portXX to your username (e.g. root+port01 or operator+port01).</p>	<pre data-bbox="675 1018 1378 1171"># SSH to serial port 1 by port name ssh -l operator+port01 70.33.235.190</pre>
<p>SSH to the Opengear node, log in adding the +port-label to your username (e.g. root+Router or operator+Router).</p>	<pre data-bbox="675 1438 1378 1591"># SSH to serial port labelled Router ssh -l operator+Router 70.33.235.190</pre>

Note:For additional reading on connecting to serial ports see:
<https://opengear.zendesk.com/hc/en-us/articles/216373543-Communicating-with-serial-port-connected-devices>

Note:Serial ports in the Local Console and Disabled ports modes are not available for SSH connection.

FEATURE PERSIST

If the node has an active console session after closing pmsHELL, connecting to the node again will resume the session and you are not prompted for the node password.

PROPERTIES AND SETTINGS

Property	Definition/Range
Serial Port Delimiter	<p>A character that separates the User name and port selection information. The default value is the + character.</p> <p><i>Default is '+', maximum length is 1.</i></p> <p><i>The prohibited characters are '\', '\"', '\'', ',', '=', and '#'.</i></p>

	<p>Source: schema</p> <p>required ssh_delimiter: string (default = "+"; minimum = 1; maximum = 1; validator = ("ssh_ url_delimiter")),</p> <p>Source: validator</p> <pre> if (strlen(v) != 1) valid = 0; else if (v[0] == "\") valid = 0; else if (v[0] == "") valid = 0; else if (v[0] == "'") valid = 0; else if (v[0] == ' ') valid = 0; // breaks sshd_config else if (v[0] == '=') valid = 0; // breaks sshd_config else if (v[0] == '#') valid = 0; // breaks sshd_config else if (!isprint(v[0])) valid = 0; else { valid = 1; } </pre>
<p>Port Number for Direct SSH Links</p>	<p>This port number will be used for direct SSH links on the serial ports page. Set this option if you have configured SSH to be reachable on a non-standard port.</p>
<p>Max Startups Start</p>	<p>The number of connections pending</p>

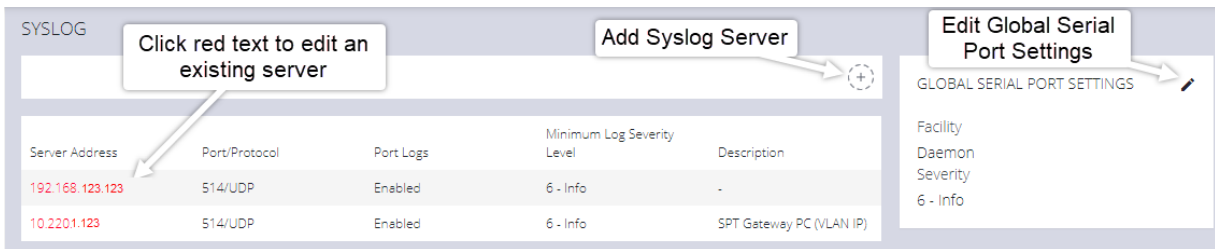
	<p>authentication before new connections <i>begin</i> to be refused.</p> <p><i>Required start: int (minimum = 1; default = 10)</i></p>
Max Startups Full	<p>The number of connections pending authentication before <i>all</i> new connections are refused.</p> <p><i>Required full: int (minimum = 1; default = 100)</i></p>
Max Startups Rate	<p>This is the percentage rate at which new connections are refused once the Max Startups value is reached. The rate is increased to 100% at Max Startup Full.</p> <p><i>Required rate: int (minimum = 1; maximum = 100; default = 30),</i></p> <p><i>The rate at which connections are refused randomly begins at max startup rate and increases linearly until the number of connections pending authentication reach max startups full, in which case 100% of new connections are refused.</i></p>
Unauthenticated Access to Serial Ports	<p>This is the feature Enable/Disable button.</p>

SYSLOG

CONFIGURE > SERVICES > Syslog

Administrative users can specify multiple external servers to which the Syslog can be exported via TCP or UDP. There is a drop-down on each serial port to enable the logging and to define the “scope” of logging.

The Syslog page lists any previously added external syslog servers.



Server Address	Port/Protocol	Port Logs	Minimum Log Severity Level	Description
192.168.123.123	514/UDP	Enabled	6 - Info	-
10.2201.123	514/UDP	Enabled	6 - Info	SPT Gateway PC (VLAN IP)

GLOBAL SERIAL PORT SETTINGS

Facility
Daemon
Severity
6 - Info

ADD A NEW SYSLOG SERVER

Note: The combination of server address, protocol and port should be unique. There can be no duplicates. However, the same server could be used if the other entry is an IPv6 address to the same Syslog server.

Use the following procedure to add a new Syslog Server.

1. Navigate to **CONFIGURE > SERVICES > Syslog**.
2. Click the **Add Syslog Server** button. The **Add Syslog Server** form opens.
3. In the **Description** field, add a suitable description that will help to identify the new server.
4. Enter the **Server Address**.
5. Click the **Protocol** switch to select either **UDP** or **TCP**.

23.10.3	FIPS Compliance	230
---------	-----------------	-----



6. Enter the correct **Port**. If no port is entered, UDP defaults to port 514 and TCP defaults to 601.
7. From the drop-down list, select the required severity level to be logged, eight levels of log severity are supported.
8. Click **Add** to complete the process.

GLOBAL SERIAL PORT SETTINGS

Global Serial Port Settings will define the Facility used and the Severity of all Syslog serial port activity sent from this node. There are two setting functions, Facility, and Severity. From the drop-down menus, select the preferred Facility and Severity as required.

GLOBAL SERIAL PORT SETTINGS TAB - FIELD DEFINITIONS

[Configure](#) > [Services](#) > [Syslog](#) > [Global Serial Port Settings](#)

Field	Definition
Description	Unique, familiar text description or name given to this syslog server that users will recognize.
Server Address	The IP address of the syslog server you are using for logging.
Protocol	Click to select the required protocol for data transmission to the syslog server.

23.10.3	FIPS Compliance	231
---------	-----------------	-----

Port	The Syslog Server IP address.
Minimum Log Severity Level	Log entries with a value equal or greater than the level specified are sent to the server.
Send Serial Port Logs	Click to enable serial port logging.
Add Button	Click to initiate the syslog, wait for confirmation banner.

SYSLOG FACILITY DEFINITIONS

Facility	Definition
Kern	Kernel messages
User	User-level messages
Mail	Mail system
Daemon	System daemons
Auth	Security/authentication messages
Syslog	Messages generated internally by syslogd
lpr	Line printer subsystem
News	Network news subsystem
uucp	UUCP subsystem
Cron	Clock daemon
Authpriv	Security/authentication messages
ftp	FTP daemon
Local	Locally used facilities

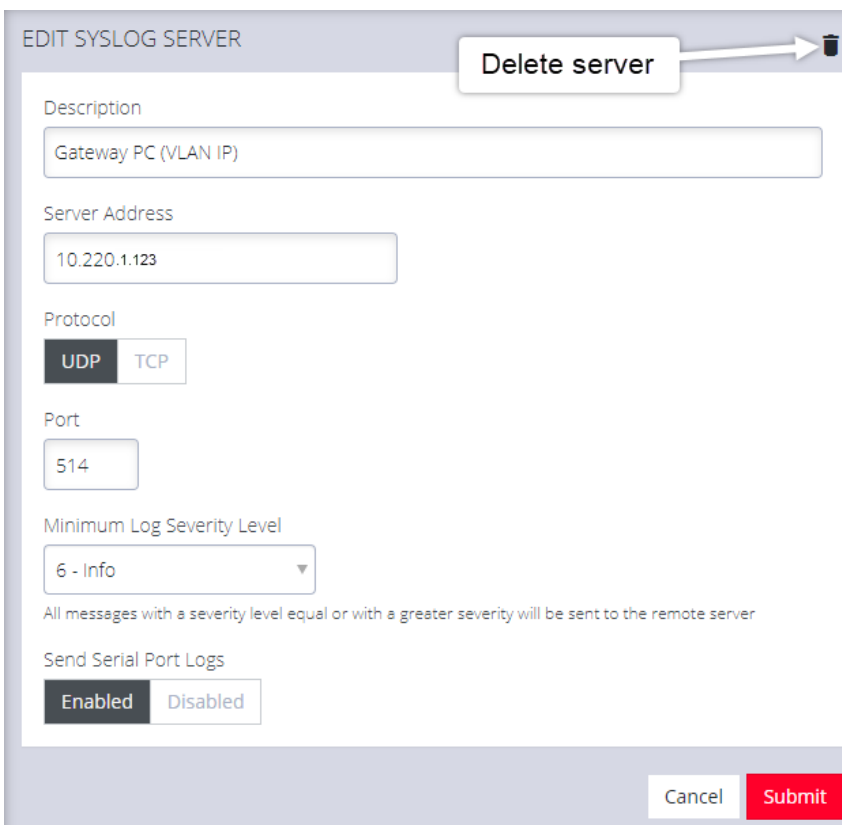
SYSLOG SEVERITY DEFINITIONS

Severity	Definition
0- Emergency	System is unusable.
1 - Alert	Action must be taken immediately.
2 - Critical	Critical conditions.
3 - Error	Error conditions.
4 - Warning	Warning conditions.
5 - Notice	Normal but significant conditions.
6 - Info	Informational messages
7- Debug	Debug-level messages

EDIT OR DELETE AN EXISTING SYSLOG SERVER

To edit an existing syslog server, click the hyperlinked **Red Text** server name in the server list (see the Syslog page image on the previous page). Make the required changes, then click the **Submit** button.

Delete a server by clicking the Delete icon at the top-right of the **Edit Syslog Server** page.



EDIT SYSLOG SERVER

Delete server

Description
Gateway PC (VLAN IP)

Server Address
10.220.1.123

Protocol
UDP TCP

Port
514

Minimum Log Severity Level
6 - Info
All messages with a severity level equal or with a greater severity will be sent to the remote server

Send Serial Port Logs
Enabled Disabled

Cancel Submit

SESSION SETTINGS

[SETTINGS](#) > [SERVICES](#) > [Session Settings](#)

23.10.3	FIPS Compliance	235
---------	-----------------	-----



Use **Session Settings** to set timeouts for console sessions where the users have been idle for a specified time. At timeout, the user's web, CLI or Serial Port sessions are terminated, thus excluding authorized users with physical access to the node that has been left connected.

To set the timeouts for Web, CLI or Serial Port sessions settings, navigate to the **SETTINGS > Services > Session Settings** page.

SESSION SETTINGS

Web Session Timeout
 minutes

CLI Session Timeout
 minutes
Set to 0 to disable.

Serial Port Session Timeout
 minutes
Set to 0 to disable.

Apply

- **Web Session Timeout:** Set the timeout from 1 to 1440 minutes.
- **CLI Session Timeout:** Set the timeout from 1 to 1440 minutes or set it to 0 to disable the timeout. Changes take effect the next time at the next login via the CLI.
- **Serial Port Session Timeout:** Set the timeout from 1 to 1440 minutes or set it to 0 to disable the timeout.

Click the **Apply** button to save the settings.

23.10.3	FIPS Compliance	236
---------	-----------------	-----



The new session timeout will take immediate effect on all pmsHELL sessions, including ones in use.

FIREWALL

CONFIGURE > FIREWALL

In the **CONFIGURE > FIREWALL** menu you can configure:

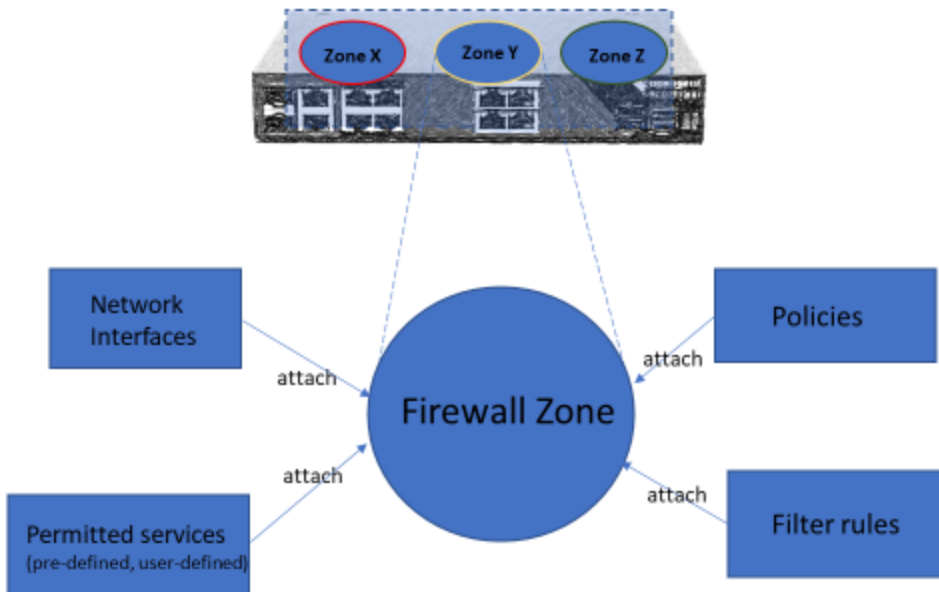
- **Firewall Management**
- **Interzone Policies**
- **Services**

23.10.3	FIPS Compliance	237
---------	-----------------	-----

FIREWALL OVERVIEW

Opengear firmware is equipped with a powerful firewall stack based on leading open source firewalld and nftables tools. The default firewall rule set is configured with a default-deny policy.

The firewall is based on the concept of configurable Zones. Zones enable operators to create multiple “firewall segments” per node and attach network interface(s), services, filtering policies and filtering rules to the zones.



Note: To access services, a user must have both access through the firewall and the appropriate authorization, e.g. via a local user account or remote AAA.

There are several kinds of rules and policies that may be applied to Zones.

Firewall Rules

- Permitted Services Rules allow access to Services for requests arriving on interfaces in the Zone – Services are configurable collections of TCP/UDP port or ports (e.g. TCP port 443 is the HTTPS service for web UI and REST API access). There are pre-defined services and supports user-defined services as well.
- Custom Rules allow the full flexibility of firewall's rich rule syntax for fine-grained access control and advanced applications.

Firewall Policies

- Interzone Policies control how Zones may forward traffic between each other – by default Zones may not forward between each other (note that interfaces in the same Zone may always forward between themselves).
- Port Forwarding Rules use destination NAT (DNAT) requests arriving on interfaces in the Zone to an external Target IP/Port, e.g. a web server running on another host
- Additionally, you can apply source NAT (SNAT) to traffic going out of a Zone by checking the Masquerade Traffic option.

EXAMPLE WEBGUI CONFIGURATION

The following examples use Permitted Services Rules and Custom Rules features

Note:Some aspects of the UI may change in future releases.

23.10.3	Firewall Overview	239
---------	-------------------	-----



Example 1: Disallow WAN Zone access to HTTPS

The default configuration is to allow HTTPS (ie. the web UI & API) on the WAN Zone. To disallow this:

Note: Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) or you will be locked out.

1. Login to the web UI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Edit Zone**.
4. Scroll down to **Permitted Services**.
5. In the right-hand column, click – to remove **https** service.
6. Any service in the right-hand column allows everyone access to this service from this zone.
7. Click **Apply**.

Example 2: Permit access to WAN Zone HTTPS from a trusted source network only

When a service is permitted using a Permitted Services Rule, connections to the service in that Zone are permitted regardless of the originating network the connection is coming from. To disallow connections from all but a trusted source network, use Custom Rules (examples below) instead.

In this example, HTTPS connections from the 10.12.34.0/24 network to the Console Manager's WAN Zone are permitted, other HTTPS connections on the WAN Zone are disallowed.

23.10.3	Firewall Overview	240
---------	-------------------	-----

Note: Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) or from the trusted source network, or you will be locked out.

1. Login to the web UI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Manage Custom Rules**.
4. Click **Add Custom Rule**.
5. In **Description** enter: *Trusted HTTPS*.
6. In **Rule Content** enter:
rule family=ipv4 source address=10.12.34.0/24 service name=https accept

Note: This is supported via firewalld 'rich-rules' option.

7. Click **Apply**.
8. Follow the steps in Example 1 above to remove the HTTPS Permitted Service.

Note: It is not recommended to mix firewall configurations between the UI (webgui/CLI) and firewalld commands (firewall-cmd) from Linux shell. Commands may be overwritten. Recommended to use either webgui or CLI for all supported functionality instead of firewall-cmd

CUSTOM RULES (FIREWALLD “RICH-RULES”)

This feature enables users to define fine-grained control of services inside a zone. Users can apply custom filter rules to traffic in a firewall zone based on Layer2 Ethernet MAC, L3 IP fields, layer 4 ports, pre-defined services. Actions to permit, deny, drop the defined packets can be included in the rule. Logging facility is also provided via custom rules.

The following sections provide examples and many sample templates that users can directly use in webGUI or CLI in the rich-rules field/section

Custom Rules Examples:

Example 1: Filter (drop) specific IPv4 source address

```
rule family="ipv4" source address="34.34.36.36" drop
```

Example 2: Permit specific source subnet and list of address

```
rule family="ipv4" source address="34.34.36.0/24" accept
```

Example 3: Permit Specific Service (HTTPS) from a specific source subnet:

```
rule family="ipv4" source address="10.0.0.0/16" service name="https"
accept
```

Example 4: Drop Specific Service (HTTP)

```
rule family="ipv4" service name="http" drop
```

Example 5: Permit specific source subnet and log connection attempts

23.10.3	Firewall Overview	242
---------	-------------------	-----

```
rule family="ipv4" source address="10.0.0.0/16" accept log
```

Example 6: Permit IPv6 packets with source address, TCP port number 4000. Log the packets

```
rule family="ipv6" source address="1:2:3:4:6::" port port=4000  
protocol=tcp accept log
```

Example 7: Permit IPv6 packets with source address, only TCP protocol, from all TCP ports. Log the packets

```
rule family="ipv6" source address="1:2:3:4:6::" protocol value="tcp"  
accept log
```


SAMPLE RICH RULES TEMPLATES

```
1. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
2. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
3. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
4. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
5. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
6. rule family="ipv4" source address="<user-to-fill>" service name="<user-to-  
fill>" accept|reject|drop
```

```
7. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
8. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
9. rule family="ipv4" source address="<user-to-fill>" port port=<usr-to-fill>  
protocol=tcp|udp accept|reject|drop
```

```
10. rule family="ipv4" source address="<user-to-fill>" protocol  
value="tcp|udp" accept|reject|drop
```

Note: The order of rules is important.

In the Template:

- Choose one of the actions accept|reject|drop [Drop action does not send any response back to source, reject does].
- For protocol value, tcp and udp examples are given in template, but many other choices are available.
- For values, source address as example, replace <user-to-fill> with the address. Address can be with or without subnet.

FIREWALL MANAGEMENT

[CONFIGURE > FIREWALL > Management](#)

Navigate to the Firewall Management page, **CONFIGURE > FIREWALL > Management**, from here you can:

- Add a new firewall zone.
- Add a firewall service.
- Edit a firewall zone - manage the zone setup.
- Manage port forwarding.
- Manage custom rules for firewalls.

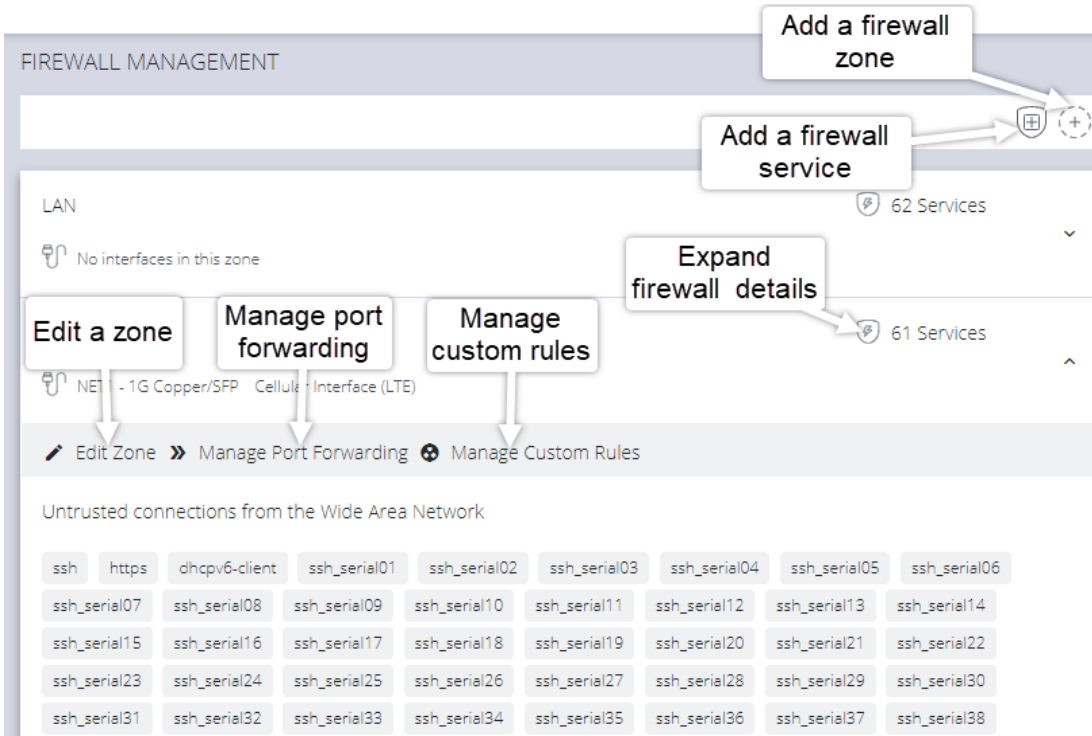


Figure: Firewall Management main page

FIREWALL ZONE SETTINGS

To change firewall management settings navigate to **CONFIGURE > FIREWALL > Management**.

Note: The application of any custom rules will result in **Permit All Traffic** being enabled in a zone.

You can inspect details of any zone by clicking the **Expand** icon to the right of the zone. Once expanded, you can click **Edit Zone** to change settings for a particular zone.

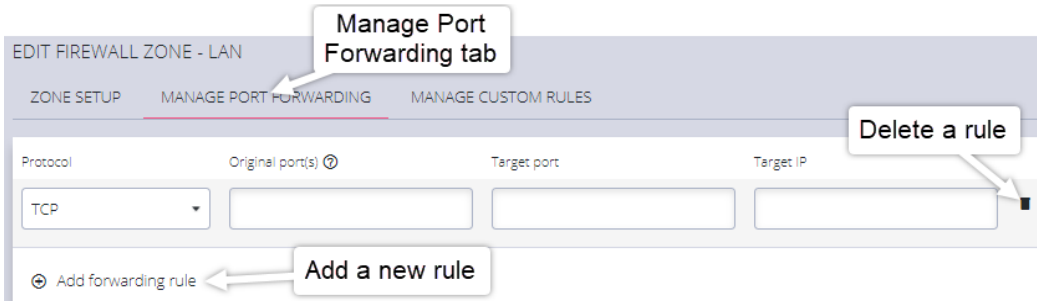
The **Edit Zone** page has three tabs. The **ZONE SETUP** page allows you to:

- Modify the Name of the zone.
- Add a Description for this zone.
- Permit all Traffic.
- Masquerade Traffic.
- Select Physical Interfaces.
- Manage Permitted Services by clicking on Plus or Minus next to each.

Tip: You can use the **Filter Interfaces** and **Filter Available Services** text boxes to limit the list content that is displayed.

PORT FORWARDING

The **MANAGE PORT FORWARDING** tab allows you to add, edit, and delete forwarding rules for the particular zone you are editing.



MANAGE CUSTOM RULES

Note: The application of any custom rules will result in **Permit All Traffic** being enabled in a zone.

The third tab, **MANAGE CUSTOM RULES**, allows you to add, edit, and delete custom firewall rules for the zone you are editing. These custom rules continue to exist after reboots, upgrades, and power cycles.

These rules are prioritized by the order they are added.

EDIT FIREWALL ZONE - LAN

ZONE SETUP

MANAGE PORT FORWARDING

MANAGE CUSTOM RULES

i All rules will be wrapped as follows:

```
firewall-cmd --permanent --zone=lan --add-rich-rule=RULE CONTENT
```

Description

Rule Content **?**

Optional description

Firewall rule - see above note on rule formatting

+ Add custom rule

Cancel

Apply

To add a new custom rule:

1. Click **Add custom rule**.
2. Enter an optional description for this rule.
3. Enter the rule content, custom rule content formatted with firewall-cmd syntax.
4. Click **Apply**.

Note:All rules will be wrapped as follows:

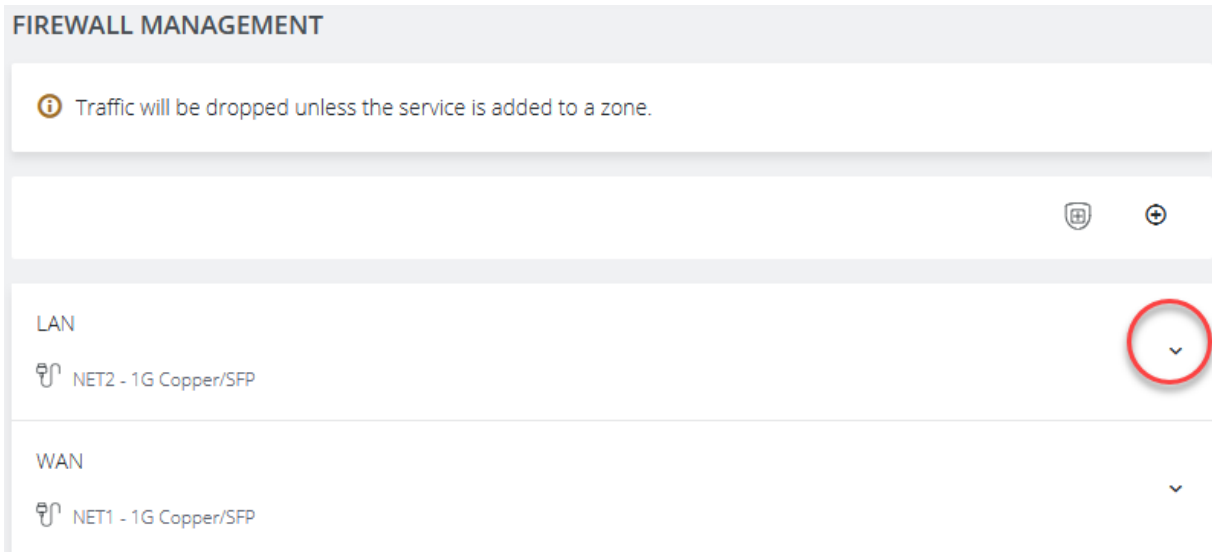
```
firewall-cmd --permanent --zone=lan --add-rich-rule=RULE CONTENT
```

FIREWALL - SOURCE ADDRESS FILTERING

Source address filtering provides an interface by which users can permit access to services (for example, SSH, HTTPS, SNMP) on a device from specific source addresses.

This feature removes generic/global permitted services within firewall zones, and instead allows users to permit a services on a specified source address (or address range) within the firewall zone. Source address filters configured in a zone apply to all the interfaces within that zone.

To access the feature, navigate to the **Configure > Firewall > Management** page through the WebUI then select the current source address filter configuration under the **services in zone** tab for each zone.




The screenshot shows the 'FIREWALL MANAGEMENT' section. At the top, there is a warning message: 'Traffic will be dropped unless the service is added to a zone.' Below this, there are two zone entries: 'LAN' with 'NET2 - 1G Copper/SFP' and 'WAN' with 'NET1 - 1G Copper/SFP'. A red circle highlights a dropdown arrow next to the LAN zone entry.

To add a source address filter for a zone, select the **edit zone** option under the desired zone, which opens the **edit zone page** where source address filters can be configured.

23.10.3	Firewall Overview	251
---------	-------------------	-----


LAN

 NET2 - 1G Copper/SFP

 [Edit Zone](#)



 [Manage Port Forwarding](#)

 [Manage Custom Rules](#)

Trusted connections from the Local Area Network

[SERVICES IN ZONE](#)

[PORT FORWARDING](#)

[CUSTOM RULES](#)

You can choose to enable permit all traffic, which will permit all traffic in the zone (unless there is a custom rule configured overwriting this behavior).

ZONE BEHAVIOR

Permit All Traffic 

Enabled

Disabled

If the permit all traffic option is disabled, you will have the option to configure permitted services for any allowed source address. Permitted services can be added or removed from each source address filter under the "Services" field.

Source address filters can be added, duplicated or deleted by using the buttons below and to the right of the filter. Any new changes to the source address filters can be seen under the **services in zone** tab for each zone on the main firewall management page.

FIREWALL SOURCE ADDRESS BULK SERVICES

[Configure](#) > [FIREWALL](#) > [Management](#) > [New Firewall Zone](#)


PERMITTED SERVICES

23.10.3	Firewall Overview	252
---------	-------------------	-----





The firewall source ip field allows you to assign permitted services to specified source ip addresses in bulk rather than needing individual rich rules to add each specific service. This change allows you to easily target specific IP Addresses with permitted services. Enter the target IP address, select services from the drop-down list and click **Apply**.


PERMITTED SERVICES

Allowed Source IP Address

(IPv4/IPv6) 

Services

<input type="text" value="insert IP address"/>	<input type="text" value="× collectd × RH-Satellite-6-capsule × amqp × apcupsd"/>	
		
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="× RH-Satellite-6-capsule × amqp × apcupsd"/>	
		

 Add a new rule

Cancel

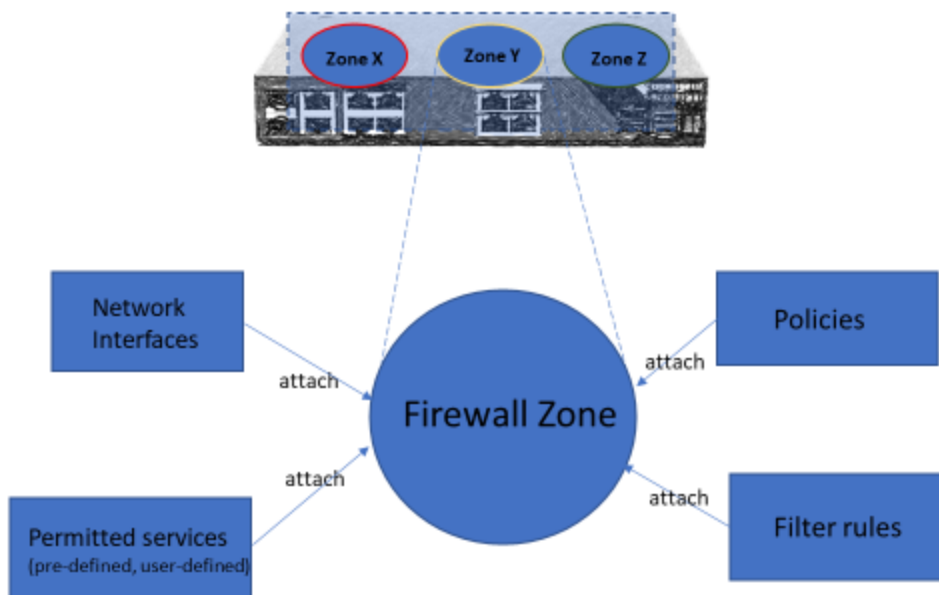
Apply

FIREWALL GUIDE

INTRODUCTION

Opengear firmware is equipped with a powerful firewall stack based on leading open source `firewalld` and `nftables` tools. The default firewall ruleset is configured with a default-deny policy.

The firewall is based on the concept of configurable Zones. Zones enable operators to create multiple “firewall segments” per node and attach network interface(s), services, filtering policies and filtering rules to the zones.



Note: To access services on the device, a user must have both access through the firewall and the appropriate authorization, e.g. via a local user account or remote AAA.



There are several kinds of rules and policies that may be applied to Zones.

Firewall Rules

- Permitted Services Rules allow access to Services for requests arriving on interfaces in the Zone – Services are configurable collections of TCP/UDP port or ports (e.g. TCP port 443 is the device's HTTPS service for WebUI and REST API access). There are pre-defined services, devices also support user-defined services.
- Custom Rules allow the full flexibility of the firewalld rich rule syntax for fine-grained access control and advanced applications.

Firewall Policies

- Interzone Policies control how Zones may forward traffic between each other – by default Zones may not forward between each other (note that interfaces in the same Zone may always forward between themselves).
- Port Forwarding Rules use destination NAT (DNAT) requests arriving on interfaces in the Zone to an external Target IP/Port, e.g. a web server running on another host
- Additionally, you can apply source NAT (SNAT) to traffic going out of a Zone by checking the Masquerade Traffic option.

EXAMPLE WEBUI CONFIGURATION

The following examples use Permitted Services Rules and Custom Rules features

Note: Some aspects of the UI may change in future releases.

23.10.3	Firewall Guide	255
---------	----------------	-----

Example 1: Disallow WAN Zone access to HTTPS

The default configuration is to allow HTTPS (i.e. the WebUI & API) on the WAN Zone. To disallow this:

Note: Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) otherwise you will lock yourself out.

1. Login to the WebUI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Edit Zone**.
4. Scroll down to **Permitted Services**.
5. In the right-hand column, click – to remove **https** service.
6. Any service in the right-hand column allows everyone access to this service from this zone.
7. Click **Apply**.

Example 2: Permit access to WAN Zone HTTPS from a trusted source network only

When a service is permitted using a Permitted Services Rule, connections to the service in that Zone are permitted regardless of the originating network the connection is coming from. To disallow connections from all but a trusted source network, use Custom Rules (examples below) instead.

In this example, HTTPS connections from the 10.12.34.0/24 network to the Operation Manager's WAN Zone are permitted, other HTTPS connections on the WAN Zone are disallowed.

Note: Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) or from the trusted source network, otherwise you will lock yourself out.

1. Login to the WebUI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Manage Custom Rules**.
4. Click **Add Custom Rule**.
5. In **Description** enter: *Trusted HTTPS*.
6. In **Rule Content** enter:
rule family=ipv4 source address=10.12.34.0/24 service name=https accept

Note: This is supported via firewalld 'rich-rules' option.

7. Click **Apply**.
8. Follow the steps in Example 1 above to remove the HTTPS Permitted Service.

Note: It is not recommended to mix firewall configurations between the UI (WebUI/CLI) and firewalld commands (firewall-cmd) from Linux shell. Commands may be overwritten. Recommended to use either WebUI or CLI for all supported functionality instead of firewall-cmd

CUSTOM RULES (FIREWALLD “RICH-RULES”)

This feature enables users to define fine-grained control of services inside a zone. Users can apply custom filter rules to traffic in a firewall zone based on Layer2 Ethernet MAC, L3 IP fields, layer 4 ports, pre-defined services. Actions to permit, deny, drop the defined packets can be included in the rule. Logging facility is also provided via custom rules.

The following sections provide examples and many sample templates that users can directly use in WebUI or CLI in the rich-rules field/section

Custom Rules Examples:

Example 1: Filter (drop) specific IPv4 source address

```
rule family="ipv4" source address="34.34.36.36" drop
```

Example 2: Permit specific source subnet and list of address

```
rule family="ipv4" source address="34.34.36.0/24" accept
```

Example 3: Permit Specific Service (HTTPS) from a specific source subnet:

```
rule family="ipv4" source address="10.0.0.0/16" service name="https"
accept
```

Example 4: Drop Specific Service (HTTP)

```
rule family="ipv4" service name="http" drop
```

Example 5: Permit specific source subnet and log connection attempts

23.10.3	Firewall Guide	258
---------	----------------	-----

```
rule family="ipv4" source address="10.0.0.0/16" accept log
```

Example 6: Permit IPv6 packets with source address, TCP port number 4000. Log the packets

```
rule family="ipv6" source address="1:2:3:4:6::" port port=4000  
protocol=tcp accept log
```

Example 7: Permit IPv6 packets with source address, only TCP protocol, from all TCP ports. Log the packets

```
rule family="ipv6" source address="1:2:3:4:6::" protocol value="tcp"  
accept log
```

USEFUL TEMPLATES FOR USE IN WEBUI OR CLI

In WebUI:

EDIT FIREWALL ZONE - BR

ZONE SETUP MANAGE PORT FORWARDING MANAGE CUSTOM RULES

ⓘ All rules will be wrapped as follows:

```
firewall-cmd --permanent --zone=br --add-rich-rule=RULE_CONTENT
```

Description ⓘ Rule Content ⓘ

Allow-34-Net rule family="ipv4" source address="34.34.36.0/24" accept

In ogcli:

```
ogcli replace firewall/zones << 'END'
```

```
firewall_zones[0].custom_rules[0].description="allow rule"
```

```
firewall_zones[0].custom_rules[0].rule_content="rule family='ipv4' source  
address='192.168.67.101/32' service name='telnet' accept"
```

...

```
END
```

SAMPLE RICH RULES TEMPLATES

```
1. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
2. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
3. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
4. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
5. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
6. rule family="ipv4" source address="<user-to-fill>" service name="<user-to-  
fill>" accept|reject|drop
```

```
7. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
8. rule family="ipv4" source address="<user-to-fill>" destination  
address="<user-to-fill>" accept|reject|drop log
```

```
9. rule family="ipv4" source address="<user-to-fill>" port port=<usr-to-fill>  
protocol=tcp|udp accept|reject|drop
```

```
10. rule family="ipv4" source address="<user-to-fill>" protocol  
value="tcp|udp" accept|reject|drop
```

Note: Ordering of rules is important. See [Firewalld Rich Rules Explained](#).

In the Template:

- Choose one of the actions `accept|reject|drop` [Drop action does not send any response back to source, reject does].
- For protocol value, `tcp` and `udp` examples are given in template, but many other choices are available.
- For values, source address as example, replace `<user-to-fill>` with the address. Address can be with or without subnet.

INTERZONE POLICIES

[CONFIGURE > FIREWALL > Interzone Policies > Create Interzone Policy](#)

In the Console Manager, Interzone firewall policy is implemented through Firewalld; this is a zone-based firewall which allows you to define zones and create rules to manage the traffic between the zones.

The firewall feature provides a dynamically managed firewall with support for network/firewall “zones” to assign a level of trust to a network and its associated connections, interfaces or sources.


The feature allows you to define policies to configure forwarding between zones and can be configured to allow directional forwarding from one or more ingress zones to one or more egress zones.

Rules and filtering may be applied at the zone level. When you add a zone, you select which services are part of that zone. Interzone policy allows these rules and filtering to be applied so as to control the type of traffic allowed to be forwarded.

The default policy, ie. when no zones are added, is that no traffic is forwarded.

CREATE AN INTERZONE POLICY

[CONFIGURE > FIREWALL > Interzone Policies > New Interzone Policy](#)

1. Navigate to the **Interzone Policies** page: [CONFIGURE > FIREWALL > Interzone Policies](#).
2. Click the **Add Firewall Policy** button  , the **New Interzone Policy** page opens for editing.
3. In the **Name** field, enter a name that clearly identifies this policy instance to other users.
4. In the **Description** field provide a detailed description of this interzone policy (optional).
5. Click to check the boxes for each Ingress and Egress zone that is to be included in this policy. You can configure traffic in both directions by selecting both zones in the Ingress and Egress as in indicated by the red arrows in the image below:

23.10.3	Firewall Guide	263
---------	----------------	-----

Two Directional Traffic Interzone Policy:

INGRESS ZONES

Traffic originating from the ingress zones will be allowed to forward to the egress zones.

Select All Zones

- LAN
- WAN
- Lighthouse VPN

EGRESS ZONES

The egress zones specify the list of zones that traffic will be forwarded to in this policy.

Select All Zones

- LAN
- WAN
- Lighthouse VPN

Cancel

Apply

Note: Additional zones may be added to the zones list at: [CONFIGURE > FIREWALL > Management > New Firewall Zone](#).

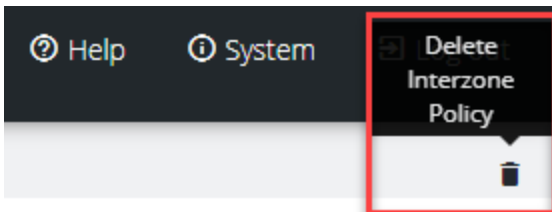
Zone customized rules may be edited at [CONFIGURE > FIREWALL > Management > Firewall Management](#).

6. Click the **Apply** button to implement the policy, a green banner will inform you that the policy details are saved successfully. The interzone policy is now active.

EDIT OR DELETE AN INTERZONE POLICY

[CONFIGURE > FIREWALL > Interzone Policies > Edit Interzone Policy](#)

1. Navigate to the **Interzone Policies** page: [CONFIGURE > FIREWALL > Interzone Policies](#).
2. Click the name of the policy you wish to edit (editable policies are identified by **red text**). The **Edit Interzone Policy** page opens for editing.
3. Edit the policy details to be changed.
4. If necessary, change the **Description** field to provide a detailed description of the edited interzone policy.
5. To **delete** a policy, click on the **Bin** widget in the top-right corner of the **Edit** page.



- 6.
7. Click the **Apply** button to implement the edited policy, a green banner will inform you that the policy details are saved successfully. The edited interzone policy is now active.

CUSTOMIZED ZONE RULES

Customized zone rules may be applied to any zone at [CONFIGURE > FIREWALL > Management > Firewall Management: "Firewall Management"](#) on page 246.

ADDING WIREGUARD ZONES TO A FIREWALL

The WireGuard interface can be added to a firewall zone as in the following example:

```
Entity firewall/zone item zone
description "" (required)
label "" (required)
masquerade "" (required)
name zone
permit_all_traffic "" (required)
address_filters (array)
custom_rules (array)
physifs (array)
port_forwarding_rules (array)
wireguards (array)
```

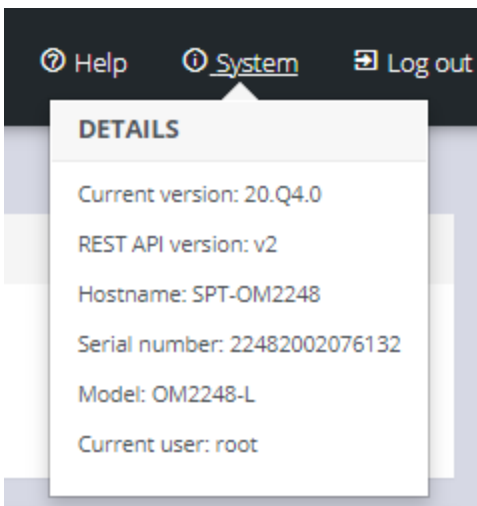
SYSTEM

CONFIGURE > SYSTEM

The **CONFIGURE > SYSTEM** menu lets you change the Console Manager hostname, perform system upgrades, and reset the system.

CHECK SYSTEM DETAILS

To ascertain current system details click on the System link at the top-right of the CM window.



ADMINISTRATION

[CONFIGURE](#) > [SYSTEM](#) > Administration

To set the hostname, add a contact email, or set a location for the Console Manager:

1. Click **CONFIGURE** > **SYSTEM** > **Administration**.
2. Edit the **Hostname** field.

ADMINISTRATION

SETTINGS

Hostname

Hostname for the system

Contact

Administration contact for the system

Location

Location for the system

Apply

3. Click **Apply**, the new settings are saved.

DATE & TIME

CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS

It is important to set the local Date and Time in your Opengear device as soon as it is configured. Features such as Syslog and NFS logging use the system time for time-stamping log entries, while certificate generation depends on a correct Timestamp to check the validity period of the certificate.

Your Opengear device can synchronize its system time with a remote Network Time Protocol (NTP) server. NTP uses Coordinated Universal Time (UTC) for all time synchronizations so it is not affected by different time zones.

You need to specify your local time zone so the system clock shows correct local time. The Date & Time section of the navigation bar provides a means to

- Set the time zone
- Manually set the correct time and date

Or

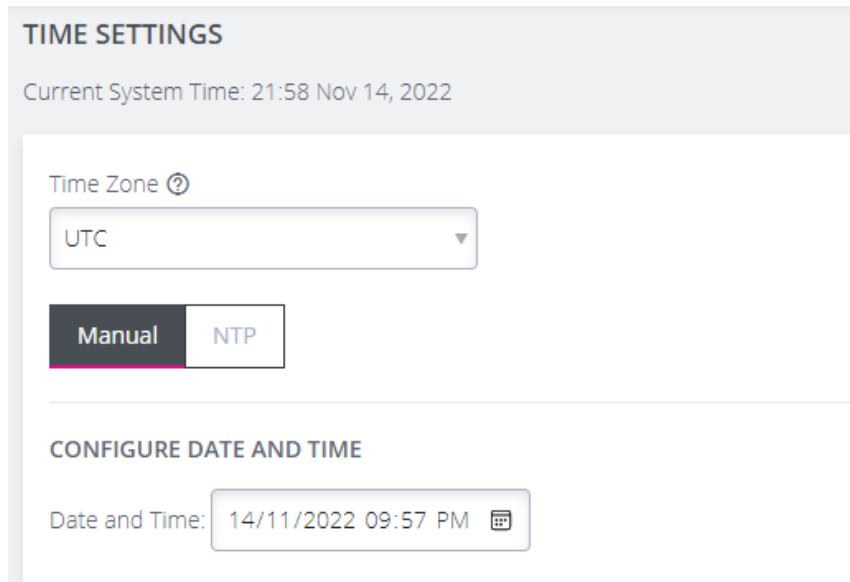
- Set the date and time by NTP Server

Continued:

23.10.3	Date & Time	269
---------	-------------	-----

MANUAL DATE & TIME SET

1. Navigate to CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS.
2. Select the applicable local time zone from the global time zone drop-down list, then, select **Manual** in the **Time Zone** section of the page.



TIME SETTINGS


Current System Time: 21:58 Nov 14, 2022

Time Zone ⓘ

UTC

Manual NTP

CONFIGURE DATE AND TIME

Date and Time: 14/11/2022 09:57 PM 

3. Select the correct date and time from the Date/Time Calendar.
4. Click the **Apply Date and Time** button.


NTP CONFIGURATION & AUTHENTICATION


Configuring an NTP server ensures the Opengear device clock is kept accurate (once Internet connection has been established).

When defining an NTP server you can choose to supply an Authentication Key and Authentication Key Identifier or not to use Authentication. If NTP Authentication keys are in use, the NTP server must be verified using the Authentication Key and Authentication Key Index before synchronizing time with the server.

23.10.3	Date & Time	270
---------	-------------	-----

1. Navigate to CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS.
2. Select the applicable time zone from the global time zone drop-down list, then, select **NTP** in the **Time Zone** section of the page.

Time Zone 

UTC 

Manual **NTP**

3. In the **Remote NTP Server List** section of the page, click **Add NTP Server**. The **'Remote NTP Server List'** opens.

Note: If your external NTP server requires authentication, you need to specify the NTP Authentication Key and the Key Index to use when authenticating with the NTP server.

REMOTE NTP SERVER LIST

NTP Server Address 

time.cloudflare.com

Authentication required

Yes No

Authentication Key 

.....

Key Index 

5

Key Format 

HEX 

Key Hash 

SHA1 

 Add NTP Server

23.10.3	Date & Time	271
---------	-------------	-----



4. Enter the IP address of the remote NTP Server.
5. If Authentication is required, select **Yes** and complete all sections of the **Authentication Key** form.
6. Click the **Apply NTP Settings** button.

CLI COMMANDS ASSOCIATED WITH NTP CONFIGURATION

Generate a new key:

```
chronyc keygen $INDEX $ALGORITHM
```

Examples:

```
chronyc keygen 1 SHA3-512
```

```
chronyc keygen 50 SHA1
```

```
chronyc keygen 2345 AES256
```

Check chronyd service:

```
systemctl status chronyd.service
```

```
journalctl -b 0 --unit chronyd.service
```

Check if the server has clients

```
chronyc clients
```

Check if the client is synchronizing:

23.10.3	Date & Time	272
---------	-------------	-----



`chronyc sources` - shows a list of servers available to the system, status, and offsets from the local clock and the source

`chronyc sourcestats` - show additional statistics for each server

`chronyc tracking` - see what server chrony is tracking with and performance metrics from that server execute

`chronyc activity` - see the number of servers and peers that are connected

`chronyc ntpdata` - returns data about each configured server

Check the NTP packets

`tcpdump -vvv -i any udp port 123`

OM specific CLI Commands

`ogcli get services/ntp`

`ogcli help services/ntp`

`ogcli replace services/ntp enabled=false` - disable NTP and clear all servers and keys.

`ogcli update services/ntp enabled=false` - disable NTP, but keep servers and keys settings.

`cat /etc/config/chronyd.conf`

`cat /etc/config/chronyd.keys`

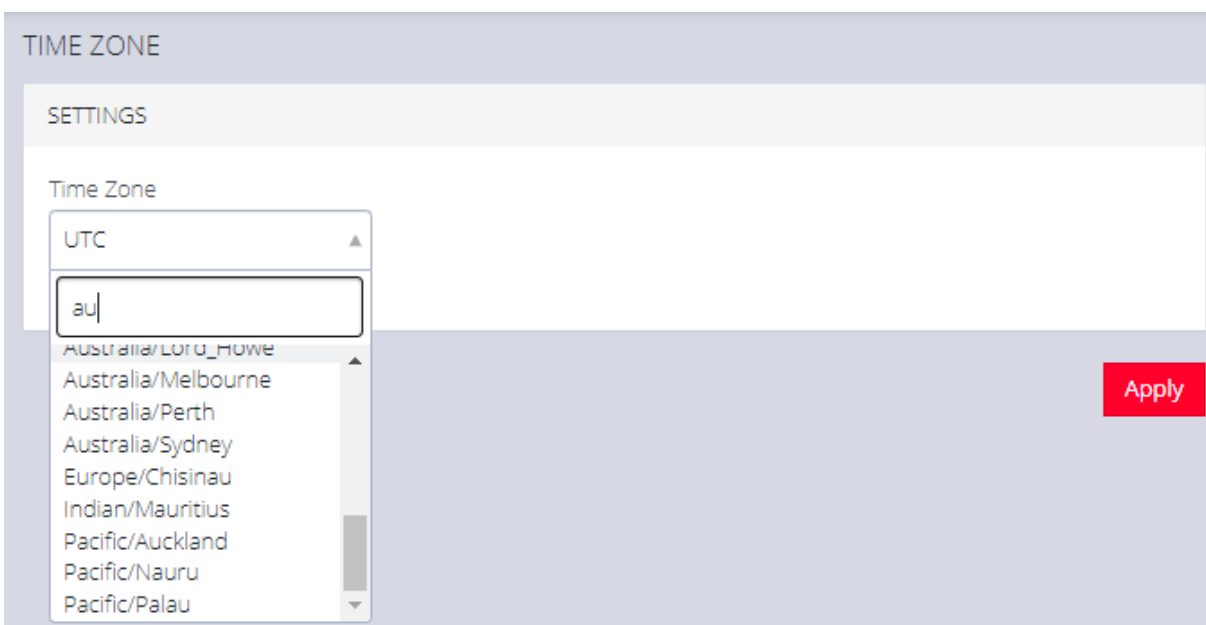
23.10.3	Date & Time	273
---------	-------------	-----

TIME ZONE

[CONFIGURE > DATE & TIME > Time Zone](#)

To set the time zone:

1. Navigate to the **CONFIGURE > DATE & TIME > Time Zone** page.
2. Select the Console Manager's time-zone from the **Time Zone** drop-down list. A filter is provided to make selection easier.
3. Click **Apply**.



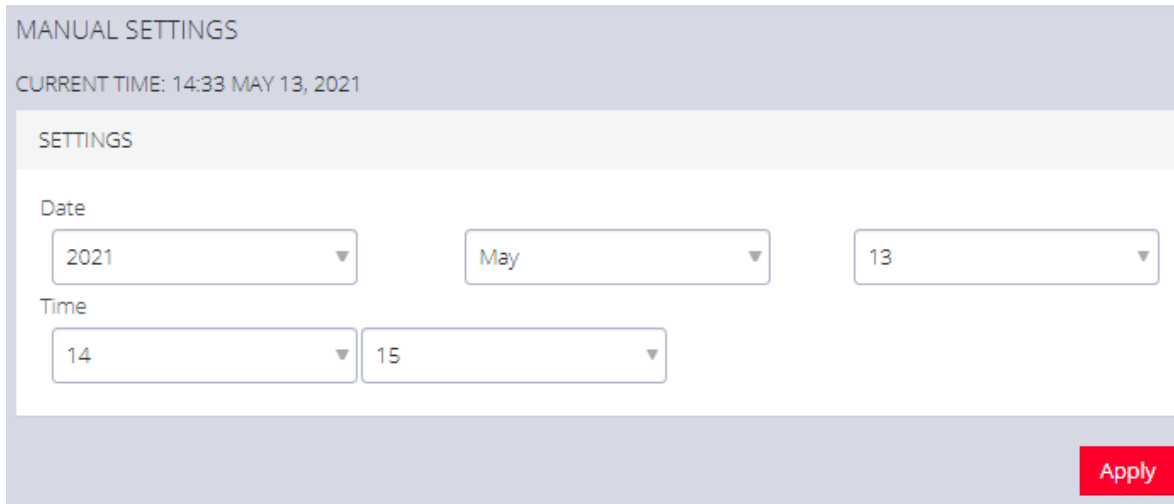
MANUAL SETTINGS

[CONFIGURE > DATE & TIME > Manual Settings](#)

To manually set the correct time and date:

23.10.3	Date & Time	274
---------	-------------	-----

1. Click **CONFIGURE > DATE & TIME > Manual Settings**.
2. Enter the current **Date** and **Time**.
3. Click **Apply**.



The screenshot shows a web interface titled "MANUAL SETTINGS". Below the title, it displays "CURRENT TIME: 14:33 MAY 13, 2021". Underneath, there is a "SETTINGS" section with two rows of dropdown menus. The "Date" row contains three dropdowns: "2021", "May", and "13". The "Time" row contains two dropdowns: "14" and "15". A red "Apply" button is located in the bottom right corner of the settings area.

Note: When Automatic time setting is enabled, the manual settings are ignored and only automatic settings are applied. Nodes enrolled in Lighthouse must be on the same time zone.

AUTOMATIC SETTINGS

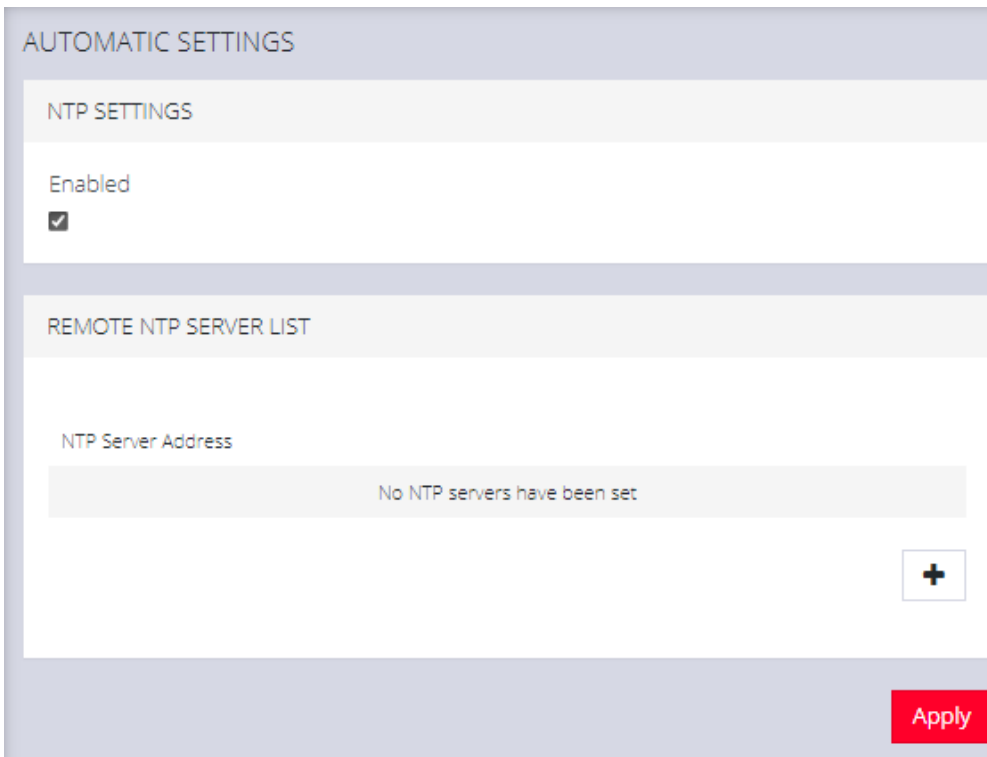
[CONFIGURE > DATE & TIME > Automatic Settings](#)

Automatic Setting of the date and time:

1. Click **CONFIGURE > DATE & TIME > Automatic Settings**.
2. Click the **Enabled** checkbox.
3. Enter a working NTP Server address in the **NTP Server Address** field.

23.10.3	Date & Time	275
---------	-------------	-----

4. Click **Apply**.



The screenshot shows the 'AUTOMATIC SETTINGS' section of the OpenGear configuration interface. It is divided into two main areas: 'NTP SETTINGS' and 'REMOTE NTP SERVER LIST'. In the 'NTP SETTINGS' section, the 'Enabled' checkbox is checked. The 'REMOTE NTP SERVER LIST' section contains a text input field labeled 'NTP Server Address' which is currently empty and displays the message 'No NTP servers have been set'. A plus sign icon is located at the bottom right of this section. A red 'Apply' button is positioned at the bottom right of the entire settings panel.

Note: When Automatic time setting is enabled, the manual settings are ignored and only automatic settings are applied. Nodes enrolled in Lighthouse must be on the same time zone.

FACTORY RESET

[CONFIGURE > SYSTEM > Factory Reset](#)

You can perform a factory reset, where logs and docker containers are preserved and everything else is reset to the factory default.

To return the Console Manager to its factory settings:

23.10.3	Date & Time	276
---------	-------------	-----



1. Select **CONFIGURE > SYSTEM > Factory Reset**.
2. Read the Factory Reset warning notice.

Warning: This will delete all configuration data from the system and reset all options to the factory defaults. Any custom data or scripts on the node will be lost. Please check the box below to confirm you wish to proceed.

3. If you still wish to proceed with the reset, Select the **Proceed with the factory reset** checkbox.
2. Click **Reset**.

Warning: This operation performs the same operation as the hard factory erase button. This resets the appliance to its factory default settings. Any modified configuration information is erased. You will be prompted to log in and must enter the default administration username and administration password (Username: root Password: default). You will be required to change this password during the first log in.

REBOOT

CONFIGURE > SYSTEM > Reboot

To reboot the Console Manager:

1. Navigate to **CONFIGURE > SYSTEM > Reboot**.
2. Select **Proceed with the reboot**,
3. Click **Reboot**.

23.10.3	Date & Time	277
---------	-------------	-----

REBOOT

WARNING

Please check the box below to confirm you wish to proceed. The appliance will reboot and will be unreachable for several minutes.

Proceed with the reboot

Reboot

EXPORT CONFIGURATION

The current system configuration can be downloaded as a plain text file. It contains all configuration performed via the Web UI and the ogcli tool.

It does not contain log files, user scripts, docker containers, service configuration or other files stored via other means.

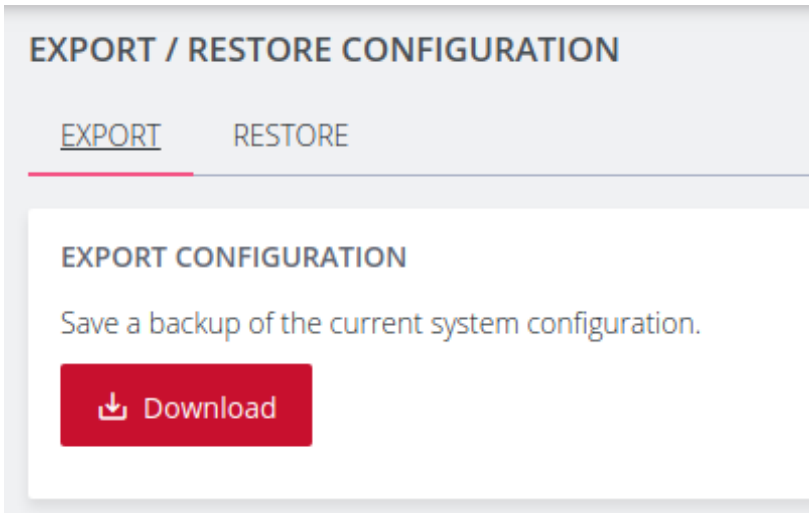
The exported configuration may be useful for:

- disaster recovery
 - issues with system upgrades
 - unexpected configuration changes
- replacing devices after RMA
- configuration templating

23.10.3	Date & Time	278
---------	-------------	-----

EXPORT CONFIGURATION VIA WEB UI

CONFIGURE > SYSTEM > Export / Restore Configuration




EXPORT / RESTORE CONFIGURATION

EXPORT RESTORE

EXPORT CONFIGURATION

Save a backup of the current system configuration.

 **Download**

To export the system configuration, click the **Download** button and save this file.

Sensitive data such as passwords and tokens will be obfuscated in the configuration export.

Note: The default filename includes the system hostname and a timestamp. For example, **cm8148_20210910_config.txt**

EXPORT CONFIGURATION VIA OGCLI

The system configuration can also be exported using the ogcli tool.

As an administrative user, run the following command:

```
ogcli export <file_path>
```

23.10.3	Date & Time	279
---------	-------------	-----

CONTROL THE EXPORT OF SENSITIVE DATA

The display of sensitive data during export via ogcli can be controlled by modifying the ogcli command:

- To display secrets in cleartext, run:

```
ogcli --secrets=cleartext export <file_path>
```

- To display obfuscated secrets, run:

```
ogcli --secrets=obfuscate export <file_path>
```

- To display secrets masked with *********, run:

```
ogcli --secrets=mask export <file_path>
```

Caution: Configuration exported with **--secrets=mask** cannot be used to import configuration.

LIGHTHOUSE NODE BACKUP

Configuration export can be scheduled to be performed periodically using the Lighthouse Node Backup feature.

For more details, consult the Lighthouse User Guide:

<https://opengear.com/support/documentation/>



RESTORE CONFIGURATION

Exported system configuration can be imported to the node using the Web UI or ogcli tool.

Note: If the configuration was exported using `--secrets=mask`, it cannot be used for configuration import.

Note: It may take up to ten minutes to import a config file with a large amount of configuration.

RESTORE CONFIGURATION VIA WEB UI

[CONFIGURE > SYSTEM > Export / Restore Configuration](#)

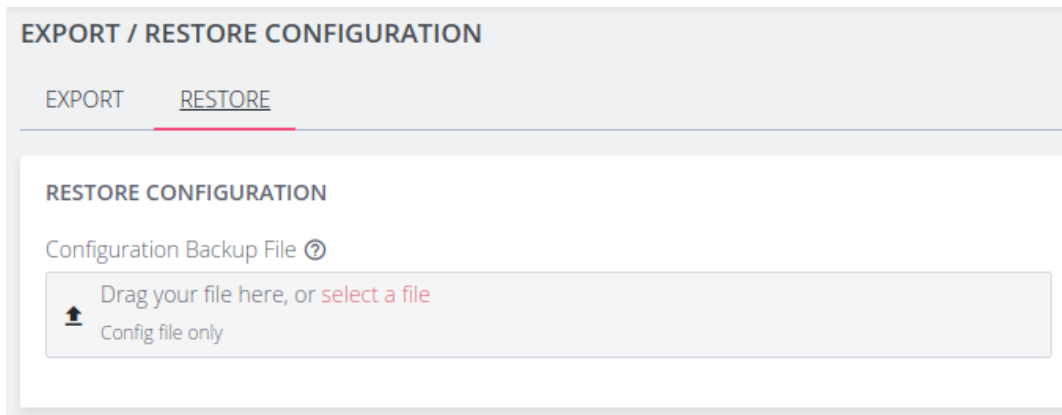
Importing configuration using the Web UI will use the restore strategy. Restoring configuration will override all settings on the node.

Only configuration from the same version and model can be restored.

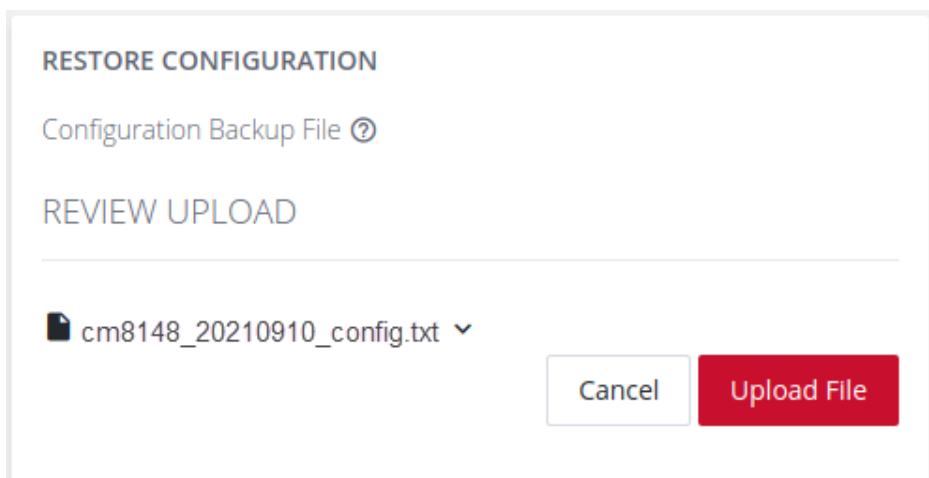
To restore the system configuration:

23.10.3	Date & Time	281
---------	-------------	-----

1. Click the **Restore** tab



2. Select the configuration file to import.
3. Review the configuration by clicking the arrow to display the file content.



4. Click the **Upload File** button to start the import process.
5. A green banner will display when the configuration import is successful.

23.10.3	Date & Time	282
---------	-------------	-----



IMPORT CONFIGURATION VIA OGCLI

The system configuration can also be imported using the `ogcli` tool. Either the import or restore strategies can be used.

IMPORT CONFIGURATION

Configuration that is imported using the `ogcli import` command will be merged with the current system configuration, preserving the current values and adding missing entries from the exported configuration where required.

As an administrative user, run the following command:

```
ogcli import <file_path>
```

RESTORE CONFIGURATION

Configuration that is imported using the `ogcli restore` command will replace the current system configuration. The resulting system configuration will reflect what is in the exported configuration.

As an administrative user, run the following command:

```
ogcli restore <file_path>
```

SYSTEM UPGRADE

[CONFIGURE > SYSTEM > System Upgrade](#)

23.10.3	Date & Time	283
---------	-------------	-----



You can perform a system upgrade when new firmware is released. After specifying the location of the firmware and beginning the upgrade process, the system will be unavailable for several minutes and then reboot. Unlike a factory reset, users, and other configuration data is maintained after the upgrade.

SYSTEM UPGRADE

SYSTEM UPGRADE

During the upgrade, the appliance will reboot and will be unreachable for several minutes.
System images must have the extension .raucb.

Upgrade Method

Fetch image from HTTP/HTTPS Server

Fetch image from HTTP/HTTPS Server

Upload image

ADVANCED OPTIONS

Upgrade Options

Only use at the request of Support

Perform Upgrade

To perform a system upgrade:

1. Navigate to the **CONFIGURE > System > System Upgrade** page.
2. Select the **Upgrade Method**, either **Fetch image from HTTP/HTTPS Server** or **Upload Image**.

Note: See <https://opengear.com/support/device-updates/> for firmware updates.

23.10.3	Date & Time	284
---------	-------------	-----



UPGRADE VIA FETCH FROM SERVER

If upgrading via **Fetch image from HTTP/HTTPS Server**:

1. Enter the URL for the system image in the **Image URL** text-entry field.
2. Click **Perform Upgrade**.

UPGRADE VIA UPLOAD

If upgrading via **Upload Image**:

1. Click the **Choose file** button.
2. Navigate to the directory containing the file.
3. Select the file and press **Return**.
4. Click **Perform Upgrade**.

Note: The **Advanced Options** section should only be used if a system upgrade is being performed as part of an OpenGear Support call.

Once the upgrade has started, the System Upgrade page displays feedback as to the state of the process.

SNMP

CONFIGURE > SNMP

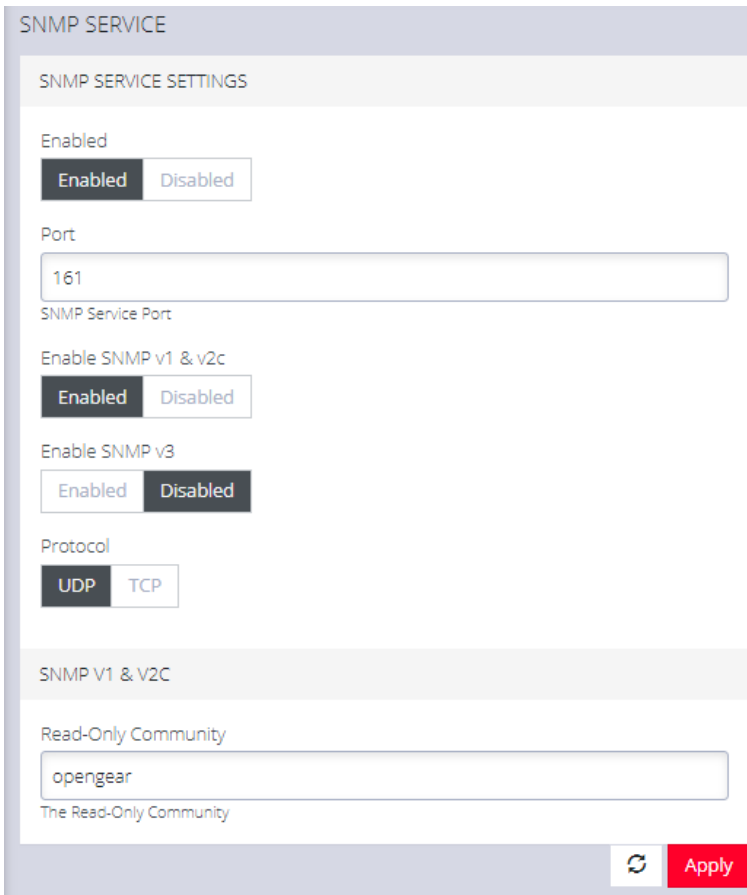
The **CONFIGURE > SNMP** menu has two options, **SNMP Service** and **SNMP Alert Managers**.

23.10.3	Date & Time	285
---------	-------------	-----

SNMP SERVICE

[CONFIGURE](#) > [SNMP](#) > [SNMP Service](#)

Navigate to the **CONFIGURE** > **SNMP** > **SNMP Service** to open the **SNMP Service** page.



SNMP Service allows you to specify which SNMP services to enable. When you click on **ENABLED** for **SNMP V1 & V2** or **SNMP V3**, a detail form appears where you can add service specific settings.

You can also specify the **SNMP Service Port** and choose between **UDP** or **TCP** for the **Protocol**.

23.10.3	Date & Time	286
---------	-------------	-----



SNMP ALERT MANAGERS

[CONFIGURE > SNMP > SNMP Alert Managers](#)

Navigate to **CONFIGURE > SNMP > SNMP Alert Managers** to open the **SNMP Alert Managers** page.

See the "[Multiple SNMP Alert Managers](#)" on the next page feature for information about configuring more than one SNMP manager.

On this page, you can set the following:

- **Address:** The IPv4 Address or domain name of the computer acting as the SNMP Manager.
- **Version:** The version of SNMP to use. The default is v2c.
- **Port:** The listening port used by the SNMP Manager. The default value is 162.
- **Manager Protocol:** The transport protocol used to deliver traps to the SNMP Manager. The default value is UDP.
- **SNMP Message Type:** The type of SNMP message to send to the SNMP manager. The INFORM option will receive an acknowledgment from the SNMP manager and will retransmit if required. The TRAP option does not expect acknowledgments.

For SNMP V1 & V2C, you can specify a **Community**. This is a group name authorized to send traps by the SNMP manager configuration for SNMP versions 1 and 2c. This must match the information that is setup in the SNMP Manager. Examples of commonly used values are log, execute, net and public.

23.10.3	Date & Time	287
---------	-------------	-----



MULTIPLE SNMP ALERT MANAGERS

[CONFIGURE > SNMP > SNMP Alert Managers > Add New SNMP Alert Manager](#)

The Multiple SNMP Alert Managers feature provides the option to configure more than one SNMP manager. Multiple SNMP Alert Managers can receive trap and inform events that can be used to trigger remedial action; events can be sent to multiple SNMP Alert Managers. The AR functionality sends traps to all configured SNMP Alert Managers for a reaction of type SNMP. Whether you input an IPv6 address or a domain name, the correct protocol needs to be selected.

CREATE OR DELETE AN SNMP MANAGER

To create a new SNMP manager:

1. Navigate to **Configure > SNMP > SNMP Alert Managers**.
2. Click the **Add New SNMP Manager** button (a plus character in the top-right of the window)
3. Complete the new **SNMP Alert Manager Form** as per the **Definitions** table below.
4. Click the **Submit** button. A banner appears confirming that the new SNMP Manager has been successfully created.
5. The new manager appears in the list of SNMP Alert Managers.
6. To delete an SNMP manager, click on the IP address of the item to open the **Edit SNMP Manager** page for that SNMP Manager.
7. Click on the **Delete SNMP Manager** widget in the top-right of the page.



Note: If you would like to use an IPv6 Address, then you need to select either UDP6 or TCP6 from the list of protocols. Whether you input an IPv6 address or a domain name, the correct protocol needs to be selected.

23.10.3	Date & Time	288
---------	-------------	-----

Note:For SNMP V3 TRAPS, an Engine ID will be provided by default if none is specified. This is generated by the snmpd service and can be found in the SNMPD RUNTIME CONF /var/lib/net-snmp/snmpd.conf. Traps will be sent for Alerts added in **Configure > SNMP Alerts**. Traps will also be sent to all the configured SNMP Alert Managers for a Playbook SNMP Reaction.

NEW SNMP ALERT MANAGER PAGE DEFINITIONS

New SNMP Alert Manager Field	Definition
Description	The editable Description field allows you to add a description of the SNMP Alert Manager.
Server Address	The IPv4/IPv6 address or domain name of the computer acting as the SNMP Alert Manager.
Port	The listening port used by the SNMP Alert Manager. The default value is 162.
Protocol	The transport protocol used to deliver traps or informs (for SNMP v3).
	UDP - Speeds up transmissions by enabling the transfer of data before an agreement is provided by the receiving party.

	<p>TCP - A commonly used protocol used to transmit data from other higher-level protocols that require all transmitted data to arrive.</p>
	<p>UDP6 - Similar to UDP but uses IPv6.</p>
	<p>TCP6 - Similar to TCP but uses IPv6.</p>
<p>Version</p>	<p>The version of SNMP protocol to use. The default value is v2c. For further reading on SNMP versions we suggest:</p> <p>https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#Protocol_versions</p>
<p>SNMP V1 & V2C Community</p>	<p>A group name authorized to send traps by the SNMP alert manager configuration for SNMP versions 1 and 2c. This will need to match what is setup in the SNMP alert manager. Examples of commonly used values are log, execute, net and public.</p>
	<p>Click the Submit button to finalize the New SNMP Manger process.</p>
	<p>Click the bin widget to Delete an SNMP Manager (in the Edit SNMP Manager page).</p>



ADVANCED OPTIONS

The Console Manager supports a number of command line interface (CLI) options and REST API.

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

external_endpoints : List of additional "address:port" endpoints to fall back to when enrolling

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

23.10.3	Advanced Options	291
---------	------------------	-----

CONFIG CLI GUIDE

The Config Command Line Interface(CLI) provides users with an interactive and familiar environment similar to other networking devices that users may be familiar with. The result is a user-experience that feels like an Interactive CLI .

Advantages of the Config CLI are:

- Interactive CLI makes everyday operations such as configuration changes and troubleshooting activities easier for users.
- Items can be created or updated without being applied immediately.
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information..
- Tab complete is supported for many commands.
- Built-in context sensitive help.
- Has a structured, tabular view when displaying lists of data.

NAVIGATION IN CONFIG CLI

STARTING A SESSION IN CONFIG CLI

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the management or local console.

EXITING A CONFIG CLI SESSION

You can exit the Interactive CLI by in any of the following ways:

- Type `exit` to end the session.
- Send an EOF (Control+D).
- Send an INT (Control+C).

Note: The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATING THE CONFIG CLI

The Config CLI operates using a hierarchy . Due to the variety of endpoints, there are several ways to get to a place where you may want to make changes.

- Starting at the root, enter endpoint names to descend down to lower endpoints.
- Similarly, type 'up' to ascend towards the root or type 'top' to reset to the root context.

Note: Every endpoint name is an operation that descends into that endpoint.

When using the config CLI, it is possible to navigate 'downwards' through multiple contexts with a single command line.

HIERARCHICAL IDENTIFIERS

This section outlines the identifiers needed to navigate the CLI.

Identifier	Description
Singleton endpoints	These require only the endpoint name to be uniquely identified.
List/item endpoints	The first level is the endpoint name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A single endpoint (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: ssh/authorized_keys > userid > [key_id]
Nested fields	The interactive CLI treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

UNDERSTANDING FIELDS, ENTITIES AND CONTEXTS

The Config CLI allows you to configure the device settings through a number of required fields, which provide the settings for the device.

These fields are grouped in *entities* that describe a small set of functionality, for example, there is a ‘user’ entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

HOW CONTEXT OPERATES IN THE CONFIG CLI

Description

The *context* is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen. Within the config shell, a number of commands are available, depending on the current context.

When config shell is started the context is at the “top context” which lists all the entities when the show command is used. If the name of an entity is typed, then the context moves 'down' into that entity. When simple commands such as `show`, `help` or `apply` are used, they will act on the current context. The context can be moved down further by typing the name of an item.

Entities can contain sub-entities as well as simple fields. For example, there is a ‘user’ entity which is used to access user settings. Fields are grouped within entities that describe a small set of functionality.

Navigating Using Context

23.10.3	Navigation in Config CLI	295
---------	--------------------------	-----

You select a context by typing the name of the target entity and pressing Enter/Return; the new context is shown in the prompt between brackets. In the following example, the 'user' context is accessed and then the 'john' sub-entity is accessed causing the context to become 'user john'.

The 'show' command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show

Item names for entity user
  john matt myuser netgrp root

config(user): john
config(user john):

Entity user item john
  description
  enabled      true
  no_password  false
  password
  ssh_password_enabled true
  groups (array)

config(user john):
```

The following example will navigate the context to the root user object without first having to navigate to the user context:

```
config: user root
config(user root):
```

Sub-objects are supported. In the following example, power_supply_voltage_alert and syslog are nested sub-objects of the onitoring/alerts/power entity:

```
config: monitoring/alerts/power power_supply_voltage_alert syslog
config(monitoring/alerts/power power_supply_voltage_alert syslog):
```

GLOBAL & ENTITY-CONTEXT COMMANDS

GLOBAL CONTEXT COMMANDS

The table below lists commands available on any context:

Global Command	Description
<code>help (or '?')</code>	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
<code>help <entity></code>	Displays short-form help for the specific entity.
<code>show</code>	Lists the available entities and fields.
<code><entity></code>	Inputting the name of an entity changes the context to focus on the named entity.
<code>exit</code>	Exit the command shell.

ENTITY CONTEXT COMMANDS

In addition to the global context commands, once an entity context is selected then further, entity context, commands become available.

Entity Command	Description
<code><field></code>	Show the value of a field.
<code>help <entity></code>	Displays short-form help for the specific entity.
<code><field> <value></code>	Set the value of a field.
<code>delete</code>	Deletes the current entity. This is available when the context entity is an item in a list.
<code>add</code>	Append a sub-entity or field to the current entity. This is only available when the context entity is a list.

CONFIG CLI ENTITIES

The config shell allows the user to configure a number of fields which are the settings for the device. These fields are grouped in entities that describe a small set of functionality. For example, there is a 'user' entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. e.g. In the following snippet, the 'user' context is accessed and then the 'john' sub-entity is accessed causing the context to become 'user john'. The 'show' command is used to list the entities and fields that descend from the current context.

SUPPORTED ENTITIES

Entity	Definition
access_right	An access right is a permit that grants the holder access to a feature or collection of related features.
auth	Configure remote authentication, authorization, accounting (AAA) servers.
auto_response/beacon	Read and manipulate the Auto-Response beacons on the NetOps Console Server appliance.

23.10.3	Config CLI Entities	300
---------	---------------------	-----

auto_response/reaction	Read and manipulate the Auto-Response reactions on the NetOps Console Server appliance.
auto_response/status	Read the AutoResponse Status on the NetOps Console Server appliance.
auto_response/status/ beacon-module	Read the AutoResponse Status of Beacon Modules on the NetOps Console Server appliance.
cellfw/info	Retrieve cellular modem version and related information.
cellmodem	Retrieve information about the cell modem.
cellmodem/sim	Cell modem SIM status.
conn	Read and manipulate the network connections on the NetOps Console Server appliance.
failover/settings	failover/settings endpoint is to check and update failover settings. When failover is enabled, this device will consume from 1MB to 1.6 MB of bandwidth per day on the probe_physif connection. If the probe addresses are unreachable, this device will take from 108 to 156 seconds to enter the failover state.
failover/status	failover/status endpoint is to check current failover status.
firewall/policy	A collection of policies defined for the NetOps Console Server appliance's firewall. A policy specifies which zones traffic is allowed to route between.
firewall/predefined_service	A collection of predefined services for the NetOps

23.10.3	Config CLI Entities	301
---------	---------------------	-----

	<p>Console Server appliance's firewall. A service is a named grouping of one or more TCP or UDP ports for a particular networking protocol. For example, the 'https' service refers to TCP port 443. This collection contains predefined services for common protocols and doesn't include the services added by the administrator.</p>
firewall/service	<p>A collection of custom services defined for the NetOps Console Server appliance's firewall. A service is a named grouping of one or more TCP or UDP ports for a particular networking protocol. For example, the 'https' service refers to TCP port 443. The appliance includes many predefined services for common protocols (see /firewall/predefined_services). This collection contains only custom services which have been defined by the administrator.</p>
firewall/zone	<p>Collection of zones defined for the NetOps Console Server appliance's firewall. A zone includes 1 or more interfaces.</p>
group	<p>Retrieve or update user group information</p>
ip_passthrough	<p>IP Passthrough endpoints are for retrieving / changing IP Passthrough settings.</p>
ip_passthrough/status	<p>The IP Passthrough status endpoint provides information about what part of the IP Passthrough connection process the device is currently at and information about the connected downstream device.</p>
ipsec_tunnel	<p>Read and manipulate the IPsec tunnels on the NetOps Console Server appliance.</p>

lighthouse_enrollment	View and control enrollment to a lighthouse.
local_password_policy	Configure the password policy for local users. This includes expiry and complexity settings.
logs/portlog	None
logs/portlog_settings	Check and update port log settings.
managementport	Used for working with local management console information
monitor/brute_force_protection/ban	Used for monitoring addresses banned by Brute Force Protection.
monitor/lldp/chassis	Get the current status of the network discovery (LLDP/CDP) protocols on this device.
monitor/lldp/neighbor	Get the list of neighboring devices (peers) that have been discovered by the LLDP protocol.
monitor/static_routes/status	Used for monitoring the status of static routes. Only IPv4 static routes are supported.
monitoring/alerts/networking	Retrieve and configure Networking Alert Group settings.
monitoring/alerts/power	Retrieve and configure Power Alert Group settings.
monitoring/alerts/system	Retrieve and configure System Alert Group settings.

pdu	Configure, monitor and control PDUs connected to the device.
pdus/drivers	Read the PDU driver list.
physif	Read and manipulate the network physical interfaces on the NetOps Console Server appliance.
port	Configuring and viewing ports information
port_session	None
ports/ auto_discover/schedule	Manage Port Auto-Discovery Scheduling
ports/status_port	Provides information about the serial pin status and Tx & Rx counters for each of this device's serial ports
system/admin_info	Retrieve or change the Operations Manager appliance system's information (hostname, contact and location)
services/ brute_force_protection	Provides access to the Brute Force Protection configuration on the system. When this service is enabled, the system watches for multiple failed login attempts and temporarily bans the offending IP Address for the configured amount of time.
services/lldp	Provides access to the Network Discovery Protocols (LLDP/CDP) configuration.
services/ntp	Provides access to the NTP client configuration on the system.

services/routing	Retrieve and configure routing services on the NetOps Console Server appliance.
services/snmp_alert_manager	SNMP Alert Managers are used to receive and log SNMP TRAP and INFORM messages sent by the NetOps Console Server. To receive SNMP alerts generated by the system at least one SNMP Alert Manager must be configured.
services/snmpd	Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behaviour. This entity allows configuration of the SNMP service.
services/ssh	Configure the Secure Shell Protocol (SSH) service.
services/syslog_server	Provides access to the remote syslog server configuration.
services/tftp	Trivial File Transfer Protocol (TFTP) is a service that allows files to be transferred to or from the NetOps Console Server appliance. This entity provides access to the TFTP server configuration on the system.
single_session	Can be enabled on a given port to prevent multiple users from connecting to that port or limit the port to a single concurrent connection.
ssh/authorized_key	Configure the SSH authorized keys for a specific user.
static_route	Configuring and viewing static routes.
system/admin_info	Retrieve or change the NetOps Console Server appli-

	ance system's information (hostname, contact and location).
system/banner	Retrieve or change the Operations Manager appliance system's banner text
system/cell_reliability_test	None
system/cellular_logging	Cellular logging provides the ability to capture the RRC connection messages from the EM7565 cellular module. This entity allows configuration of cellular logging and is only to be used during compliance testing.
system/cloud_connect	Retrieve or change the Operations Manager appliance system's cloud connect configuration
system/diskspace	Retrieve the system's Disk Space usage.
system/info	Retrieve basic system information.
system/model_name	Retrieve the Operations Manager appliance's Model Name
system/serial_number	Retrieve the Operations Manager appliance's Serial Number
system/session_timeout	Retrieve or change the Operations Manager appliance session timeouts
system/ssh_port	The SSH port used in Direct SSH links
system/ system_authorized_key	Configure the SSH authorized keys for all users.

system/time	Retrieve and update the NetOps Console Server's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the Operations Manager's most recent firmware and REST API version
user	Retrieve and update user information

CONFIG CLI COMMANDS

Command	Definition
add	Add a new item for an entity.
apply	Apply changes on just the current entity.
changes	View a list of config areas with unapplied changes.
delete	Delete an item for an entity.
discard	Discard changes on just the current entity
edit	Making changes to configuration options without navigating through the hierarchy.
exit	Leave config mode without applying changes.
help / ?	Display the available options for the configuration section. Can be used in combination with a command or configuration option to access help documentation.
import/export	Copy a config file from a specific network location to the console server and run the file. The import/export commands operate in bash, ie. outside of config CLI.

	You must exit config to operate the import/export features.
show	Display information relevant to the configuration section, highlighting changes.
up/exit/ ..	Allows users to traverse the configuration hierarchy.

ADD

Description

The `add` command will add a new item for an entity. The `add` command requires a unique value to identify the record. This will be used for the entity's label field.

The **add** command can be used:

- Anywhere within the command structure to begin the process of progressively adding an element.
- As part of a single line command where an element is added and simple fields are set.

Parameters

`entity` - the entity to which the new item will be added

`label` - a unique value to identify the record

`field` - optional field to set for the item

`value` - optional value corresponding to the field

Syntax

```
add <entity> <optional-entity> <label> <optional-field> <optional-value>
```

Example

23.10.3	Config CLI Commands	309
---------	---------------------	-----

```
add user aconsoleuser description "I am a console user"
```

APPLY

Description

The `apply` command allows users to stage configuration changes by allowing proposed changes to be held in memory, separate from active configuration until they are applied.

This may be considered from a user perspective like this:

"When I am adding users and realize that groups are missing, I can take a pause and add the groups without having to discard my work so far."

or

"When I am in the process of creating a new firewall zone but there is required service missing, I can go off and add the service and come back without losing changes."

Users can choose to apply changes in the following manner:

- Isolated changes that are specific to sections of configuration.
- Across all configurations.

Parameters

When no parameters are provided, the command will apply the changes in the current item context. For example, if the current context is `user consoleuser`, any changes to the `consoleuseruser` will be saved. If the `apply` command is used outside of an item context, this will result in an error.

`apply all` – When the 'all' parameter is added, the command will apply all changes to all items that have been changed in this session.



Syntax

```
apply [all]
```

Examples

Apply changes to a single item

These commands change a user. Then the apply command is used while still in the “user myuser” item context so only changes to this user are applied:

```
config: user myuser
config(user myuser): password secret123 description "This is my user"
config(user myuser): apply
```

Apply all changes

These commands add a new group and then change a port setting. At the end, the apply all command saves both the group and port items.

```
config: add group mygroup
config: group mygroup
config(group mygroup): access_rights
config(group mygroup access_rights): add pshell
config(group mygroup access_rights): up
config(group mygroup): ports
config(group mygroup ports): add port01
config(group mygroup ports): top
config: port port01
config(port port01): label "Port for my group"
config(port port01): top
config: apply all
```

Apply changes to specific sections of configuration

23.10.3	Config CLI Commands	311
---------	---------------------	-----

From within a specific section of hierarchy. For example, if the user is in the

```
config users johnsmith
apply
```

This will apply any changes made specifically within the user's configuration section.

Apply changes from a different section in the hierarchy

For example, if changes have been made in

```
config users johnsmith
```

but the user has moved elsewhere in the hierarchy, the command:

```
apply users johnsmith
```

will apply any changes made specifically within the user's configuration section.

Alternatively, a user might choose to apply all changes in the user list using the following command:

```
apply users
```

Using `apply` across all configurations

```
apply
```

```
apply all
```

CHANGES

Description

23.10.3	Config CLI Commands	312
---------	---------------------	-----

The `changes` command allows users to view a list of config areas with unapplied changes.

This will be a list, ordered alphabetically. Users should be able to copy and paste items from the list and use it in conjunction with the `show` command to view details.

Parameters

none

Syntax

`changes`

Examples

The following example shows changes made to multiple users and a port:

```
config: edit user root description "New description"
config: add user newuser description "New User"
config: edit port port01 baudrate 115200
config: changes

Entity user item root (edit)
  description New description
Entity user item newuser (add)
  description New User
Entity port item port01 (edit)
  baudrate 115200
```

DELETE

Description

23.10.3	Config CLI Commands	313
---------	---------------------	-----

The `delete` command is used to delete an item or entity or remove a config section or sub-section. The command requires a unique value to identify the record. This will be used for the entity's label field.

Similar to the `add` command, `delete` makes the change in a temporary state and will affect configuration only once applied.

The `delete` command can be used on:

- Existing configuration
- Unapplied changes

When used on unapplied changes, this will behave in the same way as the `discard` command.

Parameters

`entity` - the entity from which to delete the item.

`Item-label` - the label identifying the item to delete.

Syntax

```
delete <entity> <optional-entity> <item-label>
```

Example

```
delete user aconsoleuser
config:
```

Removing an element

From the users context:

```
delete "username"
```

Single line command

```
delete user "username"  
apply
```

Either of the above examples will result in exiting the context of an item being deleted.

Refer to the `apply` command for how this will behave.

DISCARD

Description

The discard command is used to remove unapplied changes.

This can be used to discard specific or configuration wide changes including:

- Updates to configuration items
- Unapplied additions
- Items designated for deletion

Parameters

`discard` - when used on its own discard the current item when in an item context, otherwise it will be an error.

`discard all` - when used with the 'all' command, then any changes staged in the current session will be dropped.

Syntax

```
discard [all]
```

Examples

The following commands create a user and then discard the user (it is never saved).

Note:The context changes to exit the 'myuser' item since it no longer exists.

```
config: add user myuser
```

```
config: user myuser
```

```
config(user myuser): discard
```

Discard changes

`config(user):`

The following commands discard changes to an existing item. The item isn't removed in this case since it has been applied previously. The description field will revert back to whatever it was before.

```
config: user root
config(user root): description "Root user"
config(user root): discard
```

The following commands discard changes to multiple entities, the group and port entities. Both will be reverted:

```
config: edit group admin description "New group description"
config: edit port port01 label "New label"
config: discard all
```

Discard all changes

```
discard *
```

This will result in a confirmation being displayed.

23.10.3	Config CLI Commands	316
---------	---------------------	-----

Discard groups of changes

```
discard auth user "username"
```

- If “username” is an addition that has not been applied, it will result in the added user being discarded. In this case the user will be prompted to confirm before the command is implemented.
- If “username” is an existing user with unapplied configuration changes, this will result in any changes there being discarded. A confirmation will be required.
- If “username” is an existing user but with no changes, the user will be informed that there are no configuration changes to discard.

Discarding specific changes

```
port port01  
discard
```

- If the entity has unapplied changes it will be discarded.
- If there are no unapplied changes an information message is displayed.

Confirmation

Discarding changes at a section, or configuration wide level will give a warning that multiple changes will be discarded.

EDIT

Description

The edit command is used when making changes to configuration options without navigating through the hierarchy.

Parameters

23.10.3	Config CLI Commands	317
---------	---------------------	-----



`entity` - the entity to be edited.

`item-label` - unique value that identifies the item.

`record field` - the field to set for the item.

`value` - the value corresponding to the field.

Syntax

```
edit <entity> <optional-entity> <item-label> <field>
```

```
<value>
```

Examples

Consider the following change to a port label:

```
config
port
port_01
label "Office-switch"
```

Alternatively, consider making the change from the root of configuration mode.

```
config
edit port port_01 label "Office-switch"
```

EXIT

Description

The `exit` command can be run at any level in the configuration structure and will allow you to leave config mode. If there are unapplied changes, you are informed and asked to confirm if you wish to proceed.

Parameters

23.10.3	Config CLI Commands	318
---------	---------------------	-----

There are no parameters applicable to the exit command.

Syntax

```
exit
```

Example

```
exit
```

HELP (OR ?)

Description

Note: Config mode will accept either `help` or a question mark `?` input.

Can be used in the following ways:

- A standalone command to view available options for the configuration section.
- In combination with a command to access help documentation.
- In combination with a configuration option to access help documentation and examples.

Parameters

The `help` command shows help for the current context.

`command` - shows help for the command.

`field` - shows help for the field.

Syntax

```
help <command or field>
```

```
<command or field> ?
```

Examples

The following will print help for the “port port01” context:

```
config(port port01): help
```

or

```
config(port port01): ?
```

The following will print help for the baudrate field when in the “port port01” context:

```
config(port port01): help baudrate
```

or

```
config(port port01): baudrate ?
```

Help command used standalone

When used by itself, `help` or `?` returns a list of available commands or configuration options.

Help used in conjunction with a command

```
apply ?
```

When used in conjunction with a command, `help` displays available sub-options.

For example, when running the `apply` command from the root config level, the `help` command notifies you that changes will traverse the configuration structure, however, when running the `help` command from within a configuration section, changes will apply to configuration options contained within.

```
add user ?
```

Displays help content including syntax and config items (mandatory and optional).

23.10.3	Config CLI Commands	321
---------	---------------------	-----

Help used with a configuration option

In the context of this example, the user is running the command from within the port configuration section and is wanting to get information on the available options.

```
pinout ?
```

This will display a list of available options.

```
label ?
```

This will display expected format and a sample.

IMPORT/EXPORT

Description

Note: The import / export and associated commands operate in bash, ie. outside of config CLI. You must exit config to operate the import/export features.

The Import / Export feature allows you to export the current configuration to a file and import or restore the configuration from that file. An import will add configuration to the current configuration and restore will replace the current configuration with the contents of the configuration file.

Import

Running the import command (within bash, not in config:) will allow you to import a configuration script from an external source file. You should point the console server to a config file on specific network location. The file will be copied to the console server and run. Depending on how it has been set up, the changes can be automatically applied after the config file is run.

Export

Running the export command (within bash, not in config:) will allow you to generate a configuration script based on the existing configuration on the console server.

This command can be run at any level in the hierarchy and used to export either:

- The configuration across the node
- Configuration specific to the users's location in the hierarchy.

```
export all current config
```

Will display all config on the console server before it has been applied for copying.

```
export all saved config
```

Will display all saved config on the console server for copying.

```
export current config
```

Will display the config from the users's current position in the navigation hierarchy.

Parameters

Import and export are run from outside of the config shell. The config command is invoked from bash with different parameters to cause it to import or export the configuration without entering the config shell.

filename – The name of the file to be imported from or exported to. If omitted then `stdin` or `stdout` will be used.

Syntax

```
config export <optional filename>
```

```
config import <optional filename>
```

Examples

```
config export /tmp/console_server.config
```

```
config import /tmp/console_server.config
```

Positional arguments

{export,import,restore,merge,replace,get}

Positional Argument	Description
export	Export the current configuration.
import	Import config from a file.
restore	Restore config from a file.
merge	Merge a provided list with existing config.
replace	Replace a list or item.
get	Display an entity's associated values
Options	

<code>-h, --help</code>	Show this help message and exit.
<code>--show-config</code>	Display the entire configuration and exit.
<code>-d</code>	Increase debugging (up to 3 times).
<code>-j</code>	Export in json format.
<code>--entities</code>	Display entities and exit.

Exporting to a file

Note: The import/export and associated commands operate in bash, ie. outside of config CLI. You must exit config to operate the import/export features.

SHOW

Description

The `show` command displays information relevant to the configuration section, including the highlighting of changes. The context in which the command is run will determine what is displayed.

At `config root`, the `show` command will display system information.

Within a config section, for example from `config > auth > user`, this will display a flat list of available users.

Parameters

--	--

show	Used on its own, will display the fields of the current context. When used in the top context, it shows the list of all entities. When used in an entity context, it shows the list of items in that entity. When used in an item context, it shows the fields and values of the current item.
entity	The entity to display, or to show details of.
item	The item to display or show details of.
field	The field to show the value of.

Syntax

show <optional entity> <optional item> <optional field>

Context

Examples using context

The following examples show how the output of the show command changes in accordance with context as it may be used at the config, physif, net1 contexts:

show - at the config context:



config: show

Entities

=====

```
access_right          pdus/drivers
auth                  physif
auto_response/beacon  port
auto_response/reaction  port_session
auto_response/status  ports/auto_discover/schedule
auto_response/status/beacon-module  ports/status_port
cellfw/info           services/brute_force_protection
cellmodem             services/lldp
cellmodem/sim         services/ntp
conn                  services/routing
failover/settings    services/snmp_alert_manager
failover/status      services/snmpd
firewall/policy       services/ssh
firewall/predefined_service  services/syslog_server
firewall/service      services/tftp
firewall/zone         ssh/authorized_key
group                 static_route
ip_passthrough        system/admin_info
ip_passthrough/status  system/banner
ipsec_tunnel          system/cell_reliability_test
lighthouse_enrollment  system/cellular_logging
local_password_policy  system/cloud_connect
logs/portlog          system/diskspace
logs/portlog_settings  system/info
managementport        system/model_name
monitor/brute_force_protection/ban  system/serial_number
monitor/lldp/chassis  system/session_timeout
monitor/lldp/neighbor  system/ssh_port
monitor/static_routes/status  system/system_authorized_key
monitoring/alerts/networking  system/time
monitoring/alerts/power  system/timezone
monitoring/alerts/system  system/version
pdu                   user
```

config:



show - at the physif context:

```
config: physif
config(physif): show
Item names for entity physif
  net1
  net2
```

```
config(physif):
```

show - at the net1 context:

```
config(physif): net1
config(physif net1): show
Entity physif item net1
  description NET1 - 1G Copper/SFP
  enabled      true
  mtu          1500
  dns (object)
    nameservers (array)
    search_domains (array)
  ethernet_setting (object)
    link_speed auto
```

```
config(physif net1):
```

Examples using parameters

The following examples show the output of the show command when used with different parameters:

23.10.3	Config CLI Commands	328
---------	---------------------	-----



```
config: show physif
Item names for entity physif
  net1
  net2

config: show physif net1
Entity physif item net1
  description NET1 - 1G Copper/SFP
  enabled      true
  mtu          1500
  dns (object)
    nameservers (array)
    search_domains (array)
  ethernet_setting (object)
    link_speed auto
```

config:

```
config: show physif net1 description
NET1 - 1G Copper/SFP
config:
```

Config

You can view the content of all configuration in JSON format.

You can also view the config of a specific section of the hierarchy you are in.

```
show-config
```

Directed Usage

You will also be able to look into a config sections using the show command. For example:

```
show auth user
```

Will display a flat list of users.

```
show auth user "username"
```

Will display the configuration for the user specified.

UP / EXIT / ..

Description

These commands allow users to traverse the configuration hierarchy.

```
up
```

The position will move one level up in the hierarchy.

If used at the root configuration level, it should point trigger the exit command.

Parameters

No parameters.

Syntax

```
up
```

```
exit
```

Examples

If, as in this example, the context is a specific port, then the ports entity can be accessed by using the `up` command then moving into another port:

```
config: port port01
config(port port01): up
config(port): port02
config(port port02):
```

HOW CHANGES ARE APPLIED OR DISCARDED

When fields and entities are changed, the changes are not immediately applied to the system configuration but remain in a staged status. Items that are staged are indicated by an '*' (asterisk) when the 'show' command is used. In addition, the 'changes' command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The 'show' command indicates the changed field with an '*'. The changes command lists the changed field.

```
config(user john): description "Admin"  
config(user john): show  
Entity user item john  
description Admin * enabled true  
no_password false password false  
password  
ssh_password_enabled true  
groups (array)
```

APPLYING OR DISCARDING CHANGES

Once fields and entities have been changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an ‘**’ when the ‘show’ command is used. In addition, the ‘changes’ command can be used to show what fields have been changed.

When any changes have been made to a single or multiple entities, the following commands become available. These commands are described in detail in the Config CLI Commands section:

Command	Description
changes	Show staged changes on all entities.
apply	Apply changes only on the current entity.
discard	Discard changes only on the current entity.
apply all	Apply changes on all entities.
discard all	Discard changes on all entities.

Example

In the following example, the user 'john' has been changed to alter the description. The 'show' command indicates the changed field with an asterisk '*'. The changes command lists the changed field.

```
config(user john): description "Scrum Master"  
config(user john): show  
Entity user item john  
description Scrum Master *  
enabled true  
no_password false  
password  
ssh_password_enabled true  
groups (array)  
config(user john): changes  
Entity user item john (edit)  
description Scrum Master  
config(user john):
```

MULTI-FIELD UPDATES

Description

Within config shell, it is possible to update multiple fields with one command line. This is restricted to 'flat' fields within the current context ie arrays and sub-objects cannot currently be updated all in one command line.

For example, the following port fields can all be changed in a single command: `baudrate`, `databits`, `escape_char`, `label`, `logging_level`, `mode`, `parity`, `pinout` and `stopbits`. Other complex fields such as `control_code` and `ip_alias` cannot be modified from the port item context in one commands (multiple commands are needed).

Example

The following command sets the `baudrate`, `escape_char` and `label` fields.

```
config(port port01): baudrate 115200 escape_char ! label "My Router"
```

The changes will be staged in config shell. Use the `apply` command to save the changes to config.

To further update the `control_codes` and `ip_aliases`, multiple commands are required as follows:

```
config(port port01): control_code
config(port port01 control_code): break b chooser c
config(port port01 control_code): up
config(port port01): ip_alias
config(port port01 ip_alias): add
config(port port01 ip_alias 1): interface net1 ipaddress 10.83.0.6/24
config(port port01 ip_alias 1): up
config(port port01 ip_alias): up
```

```
config(port port01): changes
Entity port item port01 (edit)
  control_code (object)
    break b
    chooser c
  ip_alias (array)
    1 (object)
      interface net1
      ipaddress 10.83.0.6/24
config(port port01):
```

If certain fields are hidden and only visible by first configuring other fields, these hidden fields need to be set in another line. For example, the `kernel_debug` field is only revealed by setting the field `mode` of a port to `localConsole`, so this is configured on the next line:

```
config: port port03
config(port port03): mode localConsole baudrate 115200 databits 7
label aaa
logging_level eventsOnly parity even
config(port port03): kernel_debug true
```

Error Messages

If there is an error while processing a multiple-fields command, the staged values in configuration will not be changed. If there were no staged changes on the item, then no staged changes will appear. If there were already staged changes, then those staged changes will not be affected.

In the following example, the user description was previously changed to “my user”

```
config(user consoleuser): show
Entity user item consoleuser
  description      my user *
  enabled          true
  no_password     false
  password        ""
  ssh_password_enabled true
  groups (array)
    0 consoleuser
```

If a bad field name or value is supplied on the command line, then the existing staged value is retained. The bad field name is highlighted using a ^ marker.

```
config(user consoleuser): description "My console user" invalid true
                                                                    ^
Invalid input detected at '^' marker.
config(user consoleuser):
```

If the field is missing a value, a different error message is displayed:

```
config(user consoleuser): description "My console user" enabled
Incomplete command.
config(user consoleuser): show
Entity user item consoleuser
  description      my user *
  enabled true
  no_password     false
  password        ""
```

```
ssh_password_enabled true
groups (array)
  0 consoleuser
```

The bad value for the field is indicated by an error message hinting the expected type of the value:

```
config(user consoleuser): description "My console user" enabled bad
Value bad for field enabled cannot be parsed as a boolean.
config(user consoleuser): show
Entity user item consoleuser
  description my user *
  enabled          true
  no_password      false
  password         ""
  ssh_password_enabled true
  groups (array)
    0 consoleuser
```

Changes to previous functionality

With the new `show` command, some previous syntax has changed. Just typing a field name is now an error condition. Previously this would be equivalent to the `show` command.

```
config: user root
config(user root): description
Incomplete command.
config(user root):
```

ERROR MESSAGES

When an error is made in the command line an error message which identifies the error is returned. For example, if the first token of the command is mistyped, the unknown command message is displayed.

```
config: usear root
There is no command usear root.
Type 'help' to see the available commands.
config:
config: aaaaa
There is no command aaaaa.
Type 'help' to see the available commands.
config:
```

If only the first few tokens of the command can be parsed, an error message with a ^ marker is displayed showing which part of the command cannot be parsed. If a context navigation is mistyped on the command line, then the context remains unchanged. It does not partially navigate through multiple contexts. In the following example, the context remains at the top context because `roopt` is not a valid item context in the user entity context.

```
config: user roopt
          ^
invalid input detected at '^' marker.
config:
```

STRING VALUES IN CONFIG COMMANDS

Description

The syntax for the use of string values has changed. It was previously possible to enter values containing spaces without using quotes. Multiple fields can now be assigned in one command line, quotes are required to keep field values together.

Example

The following example shows setting multiple fields where the field value for the description has spaces. The first attempt doesn't work because the second part of the description is interpreted as a field name. The second attempt is the correct syntax:

Note:In the example the syntax error in the first line is highlighted in **bold** for clarity; the correct syntax is highlighted in bold in line four.

```
config(user consoleuser): description My console user enabled true
There is no command description My console user enabled true.
Type 'help' to see the available commands.
config(user consoleuser): description "My console user" enabled true
config(user consoleuser): changes
Entity user item consoleuser (edit)
  description My console user
  enabled true
config(user consoleuser):
```

If the value itself must contain quotes, there is a triple quote form for entering string values:

```
config(user consoleuser): description ""My "console" user"" enabled true
config(user consoleuser): changes
Entity user item consoleuser (edit)
  description My "console" user
  enabled true
```

The triple quoted string is used for entering multi-line strings:

```
config(system/banner): banner """
This is a banner that has
multiple lines.
"""
config(system/banner):
```

Error Messages

If the multi-line command string cannot be tokenised, an error message will be displayed in the following form:

```
config(system/banner): banner """
aaa
""""
Invalid input. Tokens must be separated by whitespace.
Check your input and try again.
config(system/banner):
```

CONFIG CLI USE CASE EXAMPLES

ADDING A USER

The following is a fully worked example showing the adding of a new user.

Note:In the following examples, some commentary has been added, the commentary is denoted with a ‘//’ prefix. Where sessions continue onto the next page, this is shown with the comment “// session continues here:”

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for help.
// Move to the user entity

config: user
config(user): help add
Add a new item for entity user.

The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:
    Retrieve and update information for a specific user.

// Create the new user

config(user): add matt
config(user matt): show
Entity user item matt
```

```
description

// Session continues here:

enabled            true

no_password        false

password           (required)

ssh_password_enabled true

username           matt

groups (array)

// Fill out some fields

config(user matt): password topsecretpassword
config(user matt): description scrum master
config(user matt): show

Entity user item matt

  description      scrum master *
  enabled          true
  password         topsecretpassword *
  ssh_password_enabled true
  username         matt
  groups (array)

// Edit the groups

config(user matt): groups
config(user matt groups): show

Entity user item matt field groups

config(user matt groups): add // Tab completion to show available values
```

```
admin myuser netgrp

config(user matt groups): add admin

config(user matt groups): up // Exit the groups list

// Session continues here:

// Show and apply

config(user matt): show

Entity user item matt

  description      scrum master *
  enabled          true
  password         topsecretpassword *
  ssh_password_enabled true
  username         matt
  groups (array)
    0 admin *

config(user matt): apply

Creating entity user item matt.

config(user matt):
```

CONFIGURING A PORT

```
config: port

config(port): help

You are here: entity port

Description for the entity:

  Configuring and viewing ports information
```

```
Names (type <name> or help <name>)
```

```
=====
```

```
USB-A USB-E USB-front-lower port03 port07 port11 port15 port19 port23
```

```
USB-B USB-F USB-front-upper port04 port08 port12 port16 port20 port24
```

```
USB-C USB-G port01          port05 port09 port13 port17 port21
```

```
USB-D USB-H port02 port06   port10 port14 port18 port22
```

```
Commands (type help <command>)
```

```
=====
```

```
exit help show up
```

```
config(port): port01
```

```
config(port port01): baudrate // tab completion
```

```
110 1200 150 19200 230400 300 4800 57600 75
```

```
115200 134 1800 200 2400 38400 50 600 9600
```

```
config(port port01): baudrate 57600
```

```
config(port port01): label Router
```

```
config(port port01): control_code
```

```
config(port port01 control_code): break a
```

```
config(port port01 control_code): up
```

```
config(port port01): show
```

```
// Session continues here:
```

```
Entity port item port01
```

```
  baudrate          57600 *
```

```
  databits          8
```

```
  escape_char       ~
```

```
  label Router      *
```

```
  logging_level     disabled
```

```
  mode              consoleServer
```

```
parity          none
pinout          X2
stopbits        1
control_code    (object)
  break a *
  chooser
  pmhelp
  portlog
  power
  quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):
```

CONFIGURE A SINGLE SESSION ON A PORT

The feature is enabled by typing `single_session true`, then apply the change.

```
config(port port01):      single_session true
config(port port01):      apply
Updating entity port      item port01.
config(port port01):      show
Entity port item          port01
  baudrate                9600
...
single_session            true
...
_ ip_alias (array) _
```

23.10.3	Config CLI Use Case Examples	347
---------	------------------------------	-----

CONFIGURE NET1 STATIC IPV4

```
conn default-conn-1 ipv4_static_settings
  address 192.168.2.54
  gateway 192.168.2.1
top
```

CONFIGURE NET2 STATIC IPV4

```
add conn net2-static-1 mode static physif net2
conn net2-static-1 ipv4_static_settings
  address 192.168.3.58
  gateway 192.168.3.1
  netmask 255.255.255.0
top
```

CONFIGURE NET3 STATIC IPV4 FOR OM2224-24E UNITS

```
add conn net3-static-1 mode static physif net3
conn net3-static-1 ipv4_static_settings
  address 192.168.4.58
  gateway 192.168.4.1
  netmask 255.255.255.0
top
```

CONFIGURE WIREGUARD THROUGH CONFIG SHELL

WireGuard is configured through Config Shell (or REST API). The minimum configuration of WireGuard is shown in the following:

1. Provide a name for the interface (wg0 in the example below).
2. Set enabled.
3. Set the `private_key` of your WireGuard interface.
4. Add an address (at least one) for your WireGuard interface (10.0.0.1/24 in this case).
5. Add a peer with the following parameters: `endpoint_address`, `endpoint_port`, `public_key`.
6. Add an `allowed_ip` for your peer. At least one - this is the WireGuard address(es) (as it can also accept an address range) of the other interface to which you are connected.

For example:

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): private_key
AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
config(wireguard wg0): enabled true
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
config(wireguard wg0 addresses): up
config(wireguard wg0): peers
config(wireguard wg0 peers): add
config(wireguard wg0 peers 0): public_key
```

```
o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=  
config(wireguard wg0 peers 0): allowed_ips  
config(wireguard wg0 peers 0 allowed_ips): add 10.0.0.2/32  
config(wireguard wg0 peers 0 allowed_ips): up  
config(wireguard wg0 peers 0): endpoint_address 192.168.1.2  
config(wireguard wg0 peers 0): endpoint_port 51820  
config(wireguard wg0 peers 0): up  
config(wireguard wg0 peers): top
```

ROOT USER PASSWORD - CLEARTEXT

```
edit user root password newpassword
```

ROOT USER PASSWORD = PASSOWRD VIA SHA256

```
openssl passwd -5 password
```

Note:* this operation is not available in config shell

DEFINE PASSWORD COMPLEXITY RULES

```
edit local_password_policy
  password_complexity_enabled true
  password_expiry_interval_enabled true
edit local_password_policy
  password_disallow_username true
  password_must_contain_number true
  password_must_contain_special true
  password_must_contain_upper_case true
```

HOSTNAME

```
edit system/admin_info hostname "OM2216-1-lab"
```

CONTACT INFO

```
edit system/admin_info
  contact "fred.bloggs@opengear.com"
  hostname "om2216-1.lab"
  location "Happy Valley Lab"
```

TIME ZONE AND NTP

```
edit system/timezone timezone "America/New_York"  
  
edit services/ntp enabled true  
services/ntp servers  
  add  
  value "74.207.242.234"  
top
```

CREATE ADMIN USER

```
add user admin  
  description "admin"  
  enabled true  
  no_password false  
  password "password"  
  user admin groups  
  add "admin"  
top
```

CREATE BREAKGLASS USER (BELONGS TO NETGRP)

```
add user breakglass
  description "breakglass" enabled true
  no_password false
  password "password"
  user breakglass groups
  add "netgrp"
top
```

ENABLE NETGRP - SET TO CONSOLEUSER

```
edit group netgrp enabled true
group netgrp ports
  add port01
  add port02
  add port03
  add port04
top
group netgrp access_rights
  add web_ui
  add pshell
  delete admin
top
```

CHANGE SSH DELIMITTER TO : DEFAULT IS +

```
edit services/ssh ssh_url_delimiter ":"
```

CHANGE PORT LABELS

```
edit port port01 label "cisco1"  
edit port port02 label "cisco2"  
edit port port03 label "cisco3"  
edit port port04 label "cisco4"
```

ENABLE TACACS - SET MODE TO REMOTELOCAL

```
edit auth mode "tacacs"  
edit auth tacacsMethod "pap" tacacs  
Password "tac_tests"  
policy "remotelocal"  
tacacsService "raccess"  
auth tacacsAuthenticationServers  
  add  
  hostname "192.168.2.220"  
  port 49  
top
```

ENABLE LLDP ON NET1 & NET2

```
edit services/lldp enabled true  
services/lldp physifs  
  add "net1"  
  add "net2"  
top
```

ENABLE TFTP

```
edit services/tftp enabled true
```

ENABLE BOOT MESSAGES

Displays on local console port.

```
edit managementport ttyS0 kerneldebug true
```

DEFINE SESSION TIMEOUTS

```
edit system/session_timeout cli_timeout 100 serial_port_timeout 100 webui_timeout 100
```

Note:The inactivity timer starts only after you exit config shell, ie. it begins the count when you have left config and are at the bash command prompt.

DEFINE MOTD

Enter banner text within quotations.

```
edit system/banner banner ""
```

ENABLE SIMM 1 ENABLE AND ADD APN

```
edit physif wwan0 enabled true
physif wwan0 cellular_setting
    apn hologram
top
```

ENABLE SIMM 1 COMPLETE END POINTS

```
edit physif wwan0 enabled true
physif wwan0 cellular_setting
    active_sim 1
    apn hologram
    iptype IPv4v6
    sim_failback_disconnect_mode ping
    sim_failback_policy never
    sim_failover_disconnect_mode ping
    sim_failover_policy never
top
physif wwan0 cellular_setting sims 0
    fail_probe_address 8.8.8.8
    fail_probe_count 3
    fail_probe_interval 600
    fail_probe_threshold 1
    failback_delay 60
    iptype "IPv4v6"
    slot 1
top
physif wwan0 cellular_setting sims 1
    fail_probe_address 8.8.8.8
    fail_probe_count 3
    fail_probe_interval 600
    fail_probe_threshold 1
    failback_delay 60
```

```
ipype IPv4v6  
slot 2  
top
```

ENABLE FAILOVER

```
edit failover/settings enabled true probe_address 192.168.2.1 probe_physif net1
```

ADD A SYSLOG SERVER

```
services/syslog_server  
  add server1  
  address 192.168.34.113  
  protocol TCP  
  port 610  
  description "my syslog server"  
top
```

Add Five Syslog Servers

Note: Due to page width limitations, in the following example, some command lines break over two lines.

```
add services/syslog_server server0 address 192.168.34.112 min_severity notice  
port 514 port_logging_enabled true protocol UDP  
add services/syslog_server server1 address 192.168.34.113 min_severity notice  
port 514 port_logging_enabled true protocol UDP  
add services/syslog_server server2 address 192.168.34.114 min_severity notice
```

```
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server3 address 192.168.34.116 min_severity info
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server4 address 192.168.128.1 description
"lighthouse-remote-syslog" min_severity info port 514 port_logging_enabled
true protocol UDP
```

SET PORT LOGGING REMOTE SYSLOG SETTINGS

```
edit logs/portlog_settings facility daemon severity infoEnable system
monitor snmp traps
```

ENABLE SYSTEM MONITOR SNMP TRAPS

```
monitoring/alerts/power power_supply_voltage_alert
  millivolt_lower 11000
  millivolt_upper 13000
  snmp
    enabled true
  up
top
monitoring/alerts/networking cell_signal_strength_alert
  enabled true
  threshold_lower 33
  threshold_upper 66
top
monitoring/alerts/system
  authentication_alert
    enabled true
  up
  config_change_alert
    enabled true
  up
  temperature_alert
    enabled true
    threshold_lower 35
    threshold_upper 67
  up
top
```

ENABLE SNMP V2 SERVICE FOR POLLING

```
edit services/snmpd enable_legacy_versions true
enable_secure_snmp false enabled true port 161 protocol UDP
edit services/snmpd rocommunity
"TkcxJAAAABBFdSigaxdDf7whb3sxKQKnjtCuuy/0COC6rE3lUu9ghg=="
```

ENABLE 2 SNMP TRAPS AND TRAP SERVERS

Note: Due to page width limitations, in the following example, some command lines break over two lines.

```
add services/snmp_alert_manager "snmp trap server 1" address 10.1.1.199 port
162 protocol UDP version v2c
services/snmp_alert_manager "snmp trap server 1"
    community "TkcxJAAAABBFdSigaxdDf7whb3sxKQKnjtCuuy/0COC6rE3lUu9ghg==" msg_type TRAP
    top
apply all

services/snmp_alert_manager 10.1.1.199:162/UDP
    name "snmp trap server 1" privacy_password secret auth_password secret
    top
apply all
```

CREATE A STAIC ROUTE

Note:Due to page width limitations, in the following example, some command lines break over two lines.

```
add static_route "static route test" destination_address 10.0.0.0
destination_netmask 8 interface net2
```

EDIT LAN (NET2) FIREWALL ZONE

(allow only source address traffic)

```
firewall/zone lan custom_rules
  add
    description "source_net4-1"
    rule_content "rule family=ipv4 source address=192.168.3.0/24 accept"
  up
  add
    description "source_net4-2"
    rule_content "rule family=ipv4 source address=10.202.198.0/27 accept"
  up
top
```

EDIT WAN (NET1) FIREWALL ZONE

(allow only source address traffic)

```
firewall/zone wan custom_rules
  add
    description "source_net4-1"
    rule_content "rule family=ipv4 source address=192.168.2.0/24 accept"
  up
  add
    description "source_net4-2"
    rule_content "rule family=ipv4 source address=192.168.4.0/24 accept"
  up
top
```

CUSTOM_RULE EXAMPLE FOR PORT AND PROTOCOL

```
add firewall/service myports label "My Serial Ports"
firewall/service myports
  add
    port 3001
    protocol tcp
  up
  apply
top
firewall/zone wan address_filters
  add
    source_address 10.10.2.0/19
    services
      add myports
    up
  up
top
```

ENROLL INTO LIGHTHOUSE

```
add lighthouse_enrollment lh1 address 2.21.99.188 bundle om2216-1 token password
```



OPENGEAR CLI GUIDE

The **ogcli** command line tool is used for getting and setting configuration, and for retrieving device state and information. The purpose of ogcli is perform a single operation and exit. Operations are performed on a single entity, a list of entities, or all entities. Entities in ogcli are collections of related information items that represent device state, information or configuration.

For a list of operations supported by ogcli, see the "[ogcli Operations](#)" section.

Note:ogcli is not an interactive shell, it runs a single command and exits.

GETTING STARTED WITH OGCLI

The best way to get started with ogcli is to use the help command. Refer to the table below to access help topics within ogcli.

For detailed information about ogcli and how it works, view the ogcli help topic by running this command:

```
ogcli help ogcli
```

ACCESS OGCLI HELP AND USAGE INFORMATION

Help Command	Displays...
ogcli help	Basic ogcli help and usage information

Help Command	Displays...
ogcli help help	Detailed information about the help command
ogcli help operations	The full list of operations and a brief description of each
ogcli help entities	The full list of entities and a brief description of each
ogcli help syntax	How to get information into and out of ogcli
ogcli help ogcli	More detailed information about the ogcli tool
ogcli help usage	Common ogcli usage examples
ogcli help secrets	Detailed information about controlling the display of secrets in ogcli.
ogcli help <operation>	A description and example usage of a specific ogcli operation
ogcli help <entity>	A description of a specific entity and the operations it supports
ogcli help <entity> <operation>	An example of how to perform a specific operation on a specific entity

BASIC SYNTAX

The ogcli tool is always called with an operation, with most operations also taking one or more arguments specifying an entity for the operation to act on.

```
ogcli <operation> [argument] [argument]
```

OGCLI OPERATIONS

Operation	Description
get	Retrieve a list or single item
replace	Replace a list or single item
update	Update an item, supports partial edits
merge	Merge a provided list with existing config
create	Create an item
help	Display ogcli help
export	Export the system configuration

Operation	Description
import	Import system configuration, merging with current system configuration
restore	Import system configuration, replacing the current system configuration

SUPPLYING DATA TO OGCLI

For operations that modify an entity (e.g. 'update') the new information can be passed as inline positional arguments, but this quickly becomes cumbersome when setting a large number of fields. Information can instead be supplied through stdin by piping the contents of a file, or with Here Document (heredoc) style. The heredoc style is the most flexible format and is used extensively in ogcli examples.

HERE DOCUMENT

A here document (heredoc) is a form of input redirection that allows entering multiple lines of input to a command. The syntax of writing heredoc takes the following form:

```
ogcli [command] << 'DELIMITER'  
  
HEREDOC  
  
DELIMITER
```

- The first line starts with the ogcli command, followed by the special redirection operator << and a delimiting identifier. Any word can be used as the delimiter, commonly 'EOF' or 'END'.

- The `HEREDOC` block can contain multiple lines of strings, variables, commands or any other type of input. Each line can specify one field to update.
- The last line ends with the delimiting identifier used above, indicating the end of input.

```
ogcli update user <username> << 'END'
  description="operator"
  enabled=false
END
```

INLINE ARGUMENTS

Field data can be entered inline with the `ogcli` command as arguments, with each field separated by a space.

```
ogcli update user <username> enabled=false description=\"operator\"
```

PIPES AND STANDARD INPUT

The data can also be entered via `stdin` by piping the data to the `ogcli` command.

```
echo 'enabled=true description="operator"' | ogcli update user
<username>
```

Alternatively, you can provide a file via input redirection with `<`.

```
echo 'enabled=true description="operator"' > partial_record
```

```
ogcli update user <username> < partial_record
```

QUOTING STRING VALUES

All string fields require the argument to be specified with double quotes ". The shell can consume double quotes, so care must be taken when specifying strings to ensure the quotes are passed to ogcli as input.

1. Double quotes in heredoc do not need to be escaped.

```
ogcli update physif <device-identifier> << 'END'
description="test network"
END
```

2. Double quotes within single quotes do not need to be escaped.

```
ogcli update physif user <username> 'description="test user"'
```

3. Double quotes not within single quotes need to be escaped.

```
ogcli update physif user <username> description=\"test user\"
```

TAB COMPLETION

ogcli includes tab completion to assist with typing commands. When entering the start of a command, press the **<tab>** key to complete the phrase to the nearest match.

If there are multiple matches, all options will be displayed for your reference.

```
root@om1208-8e:~# ogcli get cel
cellmodem          cellmodem/sims    system/cell_reliability_test
cellfw/info        cellmodem/sims    system/cellular_logging
```

DISPLAYING SECRETS IN OGCLI

Fields containing sensitive information are called **secrets**, which are handled specially by **ogcli** to obfuscate their values when they are displayed or exported.

Passwords and private keys are examples of secret fields.

The obfuscation process provides protection against "casual observation" only and offers no cryptographic security. The **obfusc** tool can be used to obtain the clear text version of any obfuscated secret generated by any Console Manager.

For more information, view the secrets help topic by running:

```
ogcli help secrets
```

The default behavior is for secrets to be passed to ogcli in clear text, and exported or displayed in obfuscated form.

For example, setting the password:

```
ogcli update services/snmpd auth_password=\"my secret\"
```

Retrieving the password (note, the output is abridged):

```
# ogcli get services/snmpd
auth_
password="TkcxJAAAABBSB3xoFWhPA6B7sDrzq3HwaTOAO/jsURqFa0qa7hc3TA=="
```

This behavior can be overridden to display sensitive fields in clear text, obfuscated form, or masked form using the **--secrets** option. The clear text and obfuscated forms are also accepted when supplying a sensitive field.

```
# ogcli --secrets=cleartext get snmpd
auth_password="my secret"
```

```
# ogcli --secrets=obfuscate get snmpd  
auth_password="my secret"
```

```
# ogcli --secrets=mask get snmpd  
auth_password="*****"
```

If an export is performed with the **--secrets=mask** option it is impossible to subsequently import the configuration, because the secrets have been removed.



COMMON CONFIGURATION EXAMPLES

These examples contain a variety of notations and usage patterns to help illustrate the flexibility of ogcli. The examples can be copied and pasted into the CLI.

REPLACE MESSAGE OF THE DAY (MOTD) DISPLAYED AT LOGIN

```
ogcli replace banner banner=\"updated message\"
```

RETRIEVE USER RECORD

```
ogcli get user <username>
```

UPDATE ITEM WITH FIELD WHERE VALUE IS A STRING

```
ogcli update user <username> description=\"operator\"
```

UPDATE ITEM WITH FIELD WHERE VALUE IS NOT A STRING

For example, a numeric or boolean value

```
ogcli update user <username> enabled=true
```

EXPORT SYSTEM CONFIGURATION

```
ogcli export <file_path>
```

IMPORT SYSTEM CONFIGURATION

```
ogcli import <file_path>
```



RESTORE SYSTEM CONFIGURATION

```
ogcli restore <file_path>
```

COMPARE CURRENT CONFIGURATION WITH A PREVIOUSLY EXPORTED CONFIGURATION

If the template file was generated using `ogcli export <file>`, use:

```
ogcli diff <file>
```

- Config differences will be displayed in a standard diff file format.
- If the active configuration is identical to the input config a short message will indicate they are the same.

Users can also invoke a config comparison against a template file using `ogcli-diff <file>`. When using this method, the same settings are used for the comparison.

Basic help for `ogcli diff` can be accessed with `ogcli diff -h`. Detailed help for `ogcli diff` can be accessed by `ogcli help diff`. Users must have Admin rights that can access the shell to use `ogcli diff`.

ENABLE LOCAL CONSOLE BOOT MESSAGES

```
ogcli get managementports
```

```
ogcli update managementport mgmtPorts-1 kerneldebug=true
```

CREATE NEW USER

```
ogcli create user << 'END'  
description="superuser"  
enabled=true
```

```
groups[0]="admin"  
password="test123"  
username="superuser123"  
END
```

CHANGE ROOT PASSWORD

```
ogcli update user root password=\"oursecret\"
```

CREATE NEW ADMINISTRATIVE USER

```
ogcli create user << 'END'  
username="adal"  
description="Ada Lovelace"  
enabled=true  
no_password=false  
groups[0]="groups-1"  
password="oursecret"  
END
```

MANUALLY SET DATE AND TIME

```
ogcli update system/timezone timezone=\"America/New_York\"
```

```
ogcli update system/time time=\"15:30 Mar 27, 2020\"
```

ENABLE NTP SERVICE

```
ogcli update services/ntp << 'END'  
enabled=true  
servers[0].value="0.au.pool.ntp.org"
```

```
END
```

UPDATE SYSTEM HOSTNAME

```
ogcli update hostname hostname=\"system-hostname\"
```

ADJUST SESSION TIMEOUTS

```
ogcli update system/cli_session_timeout timeout=180
```

```
ogcli update system/webui_session_timeout timeout=180
```

SETUP REMOTE AUTHENTICATION WITH TACACS+

```
ogcli update auth << 'END'  
mode="tacacs"  
tacacsAuthenticationServers[0].hostname="192.168.250.21"  
tacacsMethod="pap"  
tacacsPassword="tackey"  
END
```

SETUP REMOTE AUTHENTICATION WITH RADIUS

```
ogcli update auth << 'END'  
mode="radius"  
radiusAuthenticationServers[0].hostname="192.168.250.21"  
radiusAccountingServers[0].hostname="192.168.250.21"  
radiusPassword="radkey"  
END
```

CREATE USER GROUP WITH LIMITED ACCESS TO SERIALPORTS

```
ogcli create group << 'END'  
  description="Console Operators"  
  groupname="operators"  
  role="ConsoleUser"  
  mode="scoped"  
  ports[0]="ports-10"  
  ports[1]="ports-11"  
  ports[2]="ports-12"  
END
```

VIEW AND CONFIGURE NETWORK CONNECTIONS

```
ogcli get conns
```

```
ogcli get conn system_net_conns-1
```

```
ogcli update conn system_net_conns-1 ipv4_static_  
settings.address="\192.168.0.3\"
```

```
ogcli create conn << 'END'  
  description="2nd IPv4 Static Address Example"  
  mode="static"  
  ipv4_static_settings.address="192.168.33.33"  
  ipv4_static_settings.netmask="255.255.255.0"  
  ipv4_static_settings.gateway="192.168.33.254"  
  physif="net1"  
END
```



CONFIGURE SERIAL PORTS

```
ogcli get ports
```

```
ogcli get ports | grep label
```

```
ogcli get port ports-1
```

```
ogcli update port "port05" << 'END'  
mode="consoleServer"  
label="Router"  
pinout="X2"  
baudrate="9600"  
databits="8"  
parity="none"  
stopbits="1"  
escape_char="~"  
ip_alias[0].ipaddress="192.168.33.35/24"  
ip_alias[0].interface="net1"  
logging_level="eventsOnly"  
END
```

CONFIG SHELL GUIDE

The Config Shell feature provides an interactive and familiar environment similar to older Opengear appliances. The result is a user-experience that feels like an Interactive CLI.

Advantages of the Config Shell are:

- Items can be created or updated without being applied immediately
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information.
- Tab complete is supported for many commands.
- Built-in help (see ["Global Context Commands" on page 383](#)).
- Has a structured, tabular view when displaying lists of data.

START AND END A CONFIG SHELL SESSION

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the maintenance console.

You can exit the Config Shell by any of the following:

- Type `exit` to end the session.
- Send an EOF (**Control+D**).
- Send an INT (**Control+C**).

Note: The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATE IN THE CONFIG SHELL

The Config Shell operates in a hierarchy of entities. Due to the variety of entities, there are several ways for you to get to a place where you can make changes.

Starting at the root, enter the entity names to descend down through lower entities. Every entity name is an operation that descends into that entity. Similarly, type the names of entities higher in the hierarchy to ascend towards the root.

IDENTIFIERS:

Singleton entity	Require only the entity name to be uniquely identified.
List/item entity	The first level is the entity name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A single entity (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: ssh/authorized_keys > userid > [key_id] .
Nested fields	The Config Shell treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

FIELDS, ENTITIES AND CONTEXTS

The config shell allows you to configure a number of fields which define settings.

The fields are grouped in entities that describe a small set of functionality. For example, there is a 'user' entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

CONTEXT WITHIN CONFIG SHELL

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. For example, in the following snippet, the 'user' context is accessed and then the 'john' sub-entity is accessed causing the context to become 'user john'. The 'show' command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show
Item names for entity user
  john matt myuser netgrp root
config(user): john
config(user john):
Entity user item john
  description
  enabled true
  no_password false
```

```
password
ssh_password_enabled true
groups (array)
config(user john):
```

GLOBAL CONTEXT COMMANDS

The following commands are available on any context:

help (or '?')	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
help <entity>	Show the help for the specific entity.
help <field>	Show the help for the specific field.
show	List the available entities and fields.
<entity>	Typing the name of an entity changes the context to focus on the named entity.
exit	Exit the command shell.

ENTITY CONTEXT COMMANDS

The following commands available on any entity context:

<field>	Show the value of a field.
<field> <value>	Change the field to the specific value.

delete	Delete the current entity. This is available when the context entity is an item in a list.
add	Append a sub entity or field to the current entity. This is only available when the context entity is a list.

APPLY OR DISCARD FIELD CHANGES

When fields and entities are changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an '*' when the `show` command is used. In addition, the `changes` command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The `show` command indicates the changed field with an '*'. The `changes` command lists the changed field.

```
config(user john): description "Admin"
config(user john): show
Entity user item john
  description Admin *
  enabled true
  no_password false
  password
  ssh_password_enabled true
  groups (array)
config(user john): changes
Entity user item john (edit)
  description Admin
config(user john):
```

23.10.3	Config Shell Guide	385
---------	--------------------	-----

OPERATIONS

Once a change has been made, the following commands are available:

changes	show staged changes on all entities
apply	apply changes only on the current entity
discard	discard changes only on the current entity
apply all	apply changes on all entities
discard all	discard changes on all entities

SUPPORTED ENTITIES

The following entities are supported in phase 1 of this feature and are available in release 22.06.0:

auth	Configure remote authentication, authorization, accounting (AAA) servers.
group	Retrieve or update user group information.
ip_passthrough	Passthrough entities are for retrieving / changing IP Passthrough settings.
ip_passthrough/status	The IP Passthrough status entity provides information about what part of the IP

	<p>Passthrough connection process the device is currently at and information about the connected downstream device.</p>
local_password_policy	<p>Configure the password policy for local users. This includes expiry and complexity settings.</p>
logs/portlog_settings	<p>Check and update port log settings.</p>
managementport	<p>Used for working with local management console information.</p>
port	<p>Configure and view ports information.</p>
ports/auto_discover/schedule	<p>Manage Port Auto-Discovery scheduling.</p>
system/admin_info	<p>Retrieve or change the Console Manager appliance system's information (hostname, contact and location).</p>
system/banner	<p>Retrieve or change the Console Manager appliance system's banner text.</p>
system/cloud_connect	<p>Retrieve or change the Console Manager appliance system's cloud connect configuration.</p>
system/model_name	<p>Retrieve the Console Manager appliance's Model Name.</p>
system/serial_number	<p>Retrieve the Console Manager appliance's Serial Number.</p>

system/session_timeout	Retrieve or change the Console Manager appliance session timeouts.
system/ssh_port	The SSH port used in Direct SSH links.
system/time	Retrieve and update the Console Manager's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the Console Manager's most recent firmware and REST API version.
user	Retrieve and update user information.

EXAMPLE CLI COMMANDS

ADDING A USER

In this example below, some commentary is added. Commentary added later is denoted with a `//` prefix.

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
// Move to the user entity
```

```
config: user
config(user): help add
- Add a new item for entity user. -
```

The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:

Retrieve and update information for a specific user.

```
// Create the new user
```

```
config(user): add matt
config(user matt): show
Entity user item matt
  description
  enabled true
  no_password false
  password (required)
  ssh_password_enabled true
  username matt
  groups (array)
```

```
// Fill out some fields
```

```
config(user matt): password secretpassword
config(user matt): description Admin
config(user matt): show
Entity user item matt
  description Admin *
  enabled true
  password secretpassword *
```

```
ssh_password_enabled true
username matt
groups (array)
```

```
// Edit the groups
config(user matt): groups
config(user matt groups): show
Entity user item matt field groups
config(user matt groups): add // Tab completion to show available
values
admin myuser netgrp
config(user matt groups): add admin
config(user matt groups): up // Exit the groups list
```

```
// Show and apply
config(user matt): show
Entity user item matt
description Admin *
enabled true
password secretpassword *
ssh_password_enabled true
username matt
groups (array)
0 admin *
config(user matt): apply
Creating entity user item matt.
config(user matt):
```

CONFIGURING A PORT

23.10.3	Config Shell Guide	391
---------	--------------------	-----

```

config: port
config(port): help
You are here: entity port
Description for the entity:
  Configuring and viewing ports information

Names (type <name> or help <name>)
=====
USB-A  USB-E  USB-front-lower  port03  port07  port11  port15  port19  port23
USB-B  USB-F  USB-front-upper  port04  port08  port12  port16  port20  port24
USB-C  USB-G  port01           port05  port09  port13  port17  port21
USB-D  USB-H  port02           port06  port10  port14  port18  port22

Commands (type help <command>)
=====
exit  help  show  up
config(port): port01
config(port port01): baudrate // tab completion
110   1200   150   19200  230400  300   4800   57600  75
115200 134   1800   200   2400   38400  50    600   9600
config(port port01): baudrate 57600
config(port port01): label Router
config(port port01): control_code
config(port port01 control_code): break a
config(port port01 control_code): up
config(port port01): show
Entity port item port01
  baudrate      57600      *
  databits      8
  escape_char   ~
  label         Router    *
  logging_level disabled
  mode          consoleServer
  parity        none
  pinout        X2
  stopbits      1
  control_code (object)
    break      a *
    chooser
    pnhelp
    portlog
    power
    quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):

```

ADVANCED PORTMANAGER PM SHELL GUIDE

The Portmanager program allows you to access any serial port on the console server using `pmshell` commands. It

- Routes network connection to serial ports
- Checks permissions
- Monitors and logs all the data flowing to/from the ports
- Allows you to run power commands if the serial port is associated with a PDU outlet.

RUNNING PMSHELL

`pmshell` provides an environment that allows you to access and interact with serial ports via a number of command sequences. It lets you navigate between ports using the chooser command (`~m`). For example, you can use `pmshell` to connect to port 8 via the portmanager via the following command line sequence.

```
# pmshell -l port08
```

PMSHELL COMMANDS

When running `pmshell` there are a number of command sequences that you can use that begin with the `~` key.

Note:Note: If you are connected to `pmshell` via SSH, you must add an additional `~` escape sequence.

Options	Name	Result
<code>~c</code>		The Single Session feature can be enabled or disabled by editing the <code>single_session</code> field in a given port. When a user port level administration access is logged in via <code>pmshell</code> , the port configuration menu can be accessed via any port by pressing the escape character (<code>~</code> by default) followed by <code>c</code> (<code>~c</code>).
<code>~b</code>	<code>break</code>	Generates a BREAK on the serial port (if you're doing this over ssh, you'll need to type " <code>~~b</code> ").
<code>~h</code>	<code>portlog</code>	Generates a history on the serial port. Displays the traffic logs for the port - must have port logging enabled.
<code>~.</code>	<code>quit</code>	Quits <code>pmshell</code> .
<code>~p</code>	<code>power</code>	Opens the power menu for the port. The port

Options	Name	Result
		must be configured for a PDU
~u		Opens the list of user sessions, select by number to disconnect.
~m	chooser	Connects to the port menu - go back to the serial port selection menu.
~?	pmhelp	Displays help message.

CUSTOM CONTROL CODES FOR SERIAL PORTS

Custom control codes can be defined for ease of use per port or can be applied to all ports. For example, users could define a different Power Menu control code for every port, while having a single control code for View History that applies to all ports.

Custom control codes can be used by any user with access to the serial port. In order to run the shortcuts, the user presses the CTRL key + the keycode.

Note: Only Admin users can specify short-cut control codes.

CONFIGURE CUSTOM CONTROL CODES

Admin users can configure control codes for any of the `pmshell` commands through the REST API, `ogcli` and the new interactive config shell.

Control code limitations are as follows:

- Cannot set multiple control codes for a port to use the same keycode
- The available key codes are a-z, excluding 'i' and 'm' as these can be triggered by commonly used keys TAB and BACKSPACE.

To disable a certain control code for an individual port, set the port's control code to an empty string.

CONFIGURE CONTROL CODES FOR A SPECIFIED PORT (CLI EXAMPLES)

Control Codes Action	CLI Examples
<p>Set control codes for a given port. In this example, the user sets multiple control codes for port 2</p>	<pre>ogcli update port port02 << 'END' control_code.break="b" control_code.chooser="c" control_code.pmhelp="h" control_code.portlog="l" control_code.power="p" control_code.quit="q" END</pre>
<p>Clear all control codes for a given port, in this example, port 2</p>	<pre>ogcli update port port02 << 'END' control_code.break="" control_code.chooser="" control_code.pmhelp="" control_code.portlog="" control_code.power="" control_code.quit="" END</pre>

CONFIGURE A CONTROL CODE VALUE FOR ALL PORTS

To set a particular control code to one value across all serial ports, admin users can use the script `set-serial-control-codes` from the CLI as follows:

```
set-serial-control-codes CONTROL_CODE KEY
```

where:

- **CONTROL_CODE** - Must be one of the following values: `break`, `chooser`, `pmhelp`, `portlog`, `power` or `quit`.
- **KEY** - Must be a single lower case letter a-z excluding 'i' and 'm' or an empty string designated by '' which is used to clear the control code.

CONTROL CODES FOR ALL PORTS VIA CLI (EXAMPLES)

Control Codes Action	CLI Examples
Set chooser control code to CTRL-a on all ports	<pre>set-serial-control-codes chooser a</pre>
Clear chooser control code on all ports	<pre>set-serial-control-codes chooser ''</pre>

DOCKER

Docker is a tool designed to make it easier to create, deploy, and run applications by distributing them in containers. Developers can use containers to package up an application with all of the parts it needs, like libraries and dependencies, and then ship it out as one package. Docker is running by default on the Console Manager. You can access commands by typing `docker` in the Local Terminal or SSH.

For more information on Docker, enter `docker --help`.

CRON

Cron service can be used for scheduled cron jobs runs. Daemon can be managed via the `/etc/init.d/crond` interface, and cron tables managed via `crontab`. `Crontab` supports:

Usage:

`crontab [options] file`

`crontab [options]`

`crontab -n [hostname]`

OPTIONS:

`-u <user>` define user

`-e` edit user's crontab

`-l` list user's crontab

`-r` delete user's crontab



- i prompt before deleting
- n <host> set host in cluster to run users' crontabs
- c get host in cluster to run users' crontabs
- x <mask> enable debugging

To perform start/stop/restart on crond service:

```
/etc/init.d/crond start
```

Cron doesn't need to be restarted when crontab file is modified, it examines the modification time on all crontabs and reload those which have changed.

To verify the current crond status:

```
/etc/init.d/crond status
```

To check current cron jobs running with the following command to list all crontabs:

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

This opens a personal cron configuration file. Each line can be defined as one command to run. The following format is used:

minute hour day-of-month month day-of-week command

For example, append the following entry to run a script every day at 3 am:

```
0 3 * * * /etc/config/backup.sh
```

Save and close the file.

23.10.3	Custom Control Codes for Serial Ports	400
---------	---------------------------------------	-----

INITIAL PROVISIONING VIA USB KEY

Also known as “ZTP over USB”, this feature allows provisioning an unconfigured (factory erased) unit from a USB storage device like a thumb drive.

The USB device must contain a filesystem recognized by the CM (currently FAT32 or ext4) with a file named manifest.og in the root directory. This file specifies which provisioning steps will be done. An article with a partial description of the file format is here:

<https://opengear.zendesk.com/hc/en-us/articles/115002786366-Automated-enrollment-using-USB>

The USB device can be inserted any time (before or after power is applied to the unit) and as long as the unit is unconfigured, the ZTP over USB process will be triggered. Here “unconfigured” has the same meaning as for ZTP: no changes made to the ogconfig data store.

Note: Setting the root password on first log in counts as a config change.

The following manifest.og keys are implemented. This provides image installation, Lighthouse enrollment, and arbitrary script execution:

manifest.og contains <key>=<value> pairs. Recognized keys are:

image : Firmware image file name on the USB device's filesystem that will be flashed after boot once the image is validated

script : Configuration script to run

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment



external_endpoints : List of additional "address:port" endpoints to fall back to when enrolling

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle











EULA AND GPL




The current Opengear End-User License Agreement and the GPL can be found at <http://opengear.com/eula>.

23.10.3	Custom Control Codes for Serial Ports	402
---------	---------------------------------------	-----

UI BUTTON DEFINITIONS

The table below provides a definition of the button icons used in the UI.

Button Icon	Definition
	Edit buttons
	Add item (eg. SNMP Manager)
 	VLAN interface or create VLAN interface.
 	Bonded interfaces or create new bond
 	Bridged interfaces or create new bridge
	Standard network interface
	Cellular interface

 A green icon representing a bridge connection between two nodes.	Interface with bridge
 A green icon representing a bond connection between two nodes.	Interface with bond
 A grey icon of a trash bin.	Bin widget. Delete selected object.