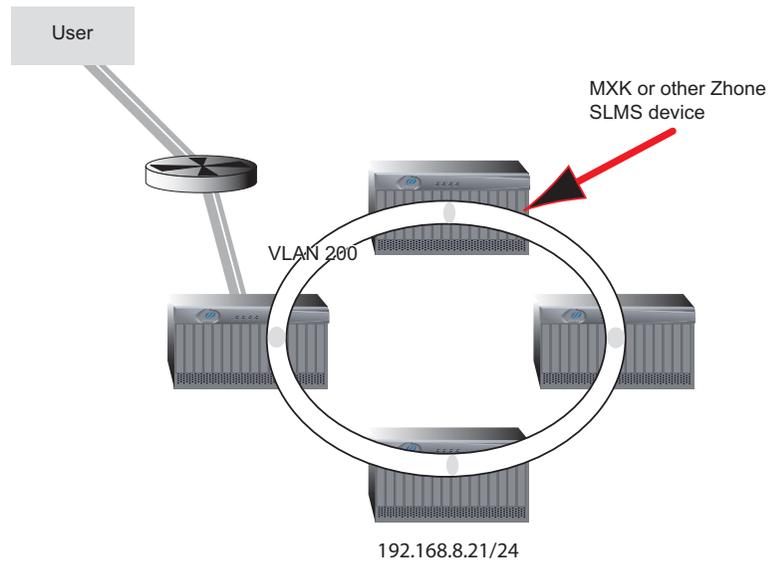


**Figure 3: IP on a bridge**



### **Configure an IP address on a Ethernet uplink port for MXK in-band management**

Configure an IP interface on an uplink port for in-band MXK management.

### **Configure IP on a bridge for Ethernet**

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24, and a logical port interface 6 on VLAN 200.

### **Creating IP on a bridge on a uplink bridge for Ethernet**

- 1 Create an *uplink* bridge with a VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl   Tagged 200  1/a/2/0/eth  ethernet2-200/bridge  UP  S VLAN 200 default
1 Bridge Interfaces displayed
```

- 2 Enter **interface add interface/type** with the type as *ipobridge*.

This command creates the new IP interface as well as the new *ipobdwn* bridge. The bridge created will be a subscriber facing downlink bridge.



**Note:** The logical port interface for IP on a bridge on the MXK must be *1-a-6-0/ipobridge* for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 200 192.168.8.21/24
Created ip-interface-record ipobridge-200/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN ID.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

### 3 Verify the *ipobridge* interface:

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
```

### 4 Verify the *ipobridge* and the *uplink* bridge:

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
upl   Tagged 200    1/a/2/0/eth       ethernet2-200/bridge  UP S VLAN 200 default
ipobdnw  Tagged 200    1/b/6/0/ipobridge ipobridge-200/bridge  UP S 00:01:47:93:74:54
                                           S 192.168.8.21
```

2 Bridge Interfaces displayed

The downlink bridge with the same VLAN ID was automatically created.

### 5 Create the default route.

See [Creating a default route on page 57](#).

## Deleting the IP on a bridge management interface

### 1 View the IP interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
```

### 2 Delete the *ipobridge* interface.

```
zSH> interface delete 1/a/6/0/ip
Delete complete
```

This action automatically deletes the *ipobridge downlink* bridge.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
upl           Tagged 200   1/a/2/0/eth      ethernet2-200/bridge  UP  S VLAN 200 default
1 Bridge Interfaces displayed
```

### 3 Delete the *uplink* bridge.

```
zSH> bridge delete ethernet2-200/bridge vlan 200
Bridge-path deleted successfully
ethernet2-200/bridge delete complete
```

## Configure TLS IP on a bridge

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24 on VLAN 200.

### Creating IP on a bridge for a TLS bridge

#### 1 Create a *tls* bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth tls vlan 700
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls           700       1/a/2/0/eth      ethernet2/bridge  UP
1 Bridge Interfaces displayed
```

#### 2 Enter **interface add** *interface/type* with the type as *ipobridge*.

This command creates the new IP interface as well as a new bridge. The bridge created will be a Transparent Lan Service (TLS) tagged bridge.



**Note:** The logical port interface for IP on a bridge on the MXK *must* be *1-a-6-0/ipobridge* for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 700 192.168.8.21/24
Created ip-interface-record ipobridge-700/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

**3** Verify the *ipobridge* interface:

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-700
-----
```

**4** Verify the *tls* IP on an bridge interface.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
tls           700        1/a/2/0/eth       ethernet2/bridge   UP
ipobtls      Tagged 700      1/a/6/0/ipobridge ipobridge-700/bridge UP S 00:01:47:93:74:54
                                           S 192.168.8.21
```

2 Bridge Interfaces displayed

The *ipobridge* creates a static IP address and MAC address.

```
zSH> bridge-path show
VLAN/SLAN  Bridge              Address
-----
700 N/A                                         VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0,
IGMP DSCP: 0, Flap Mode: Fast
700 ipobridge-700/bridge 192.168.8.21
700 ipobridge-700/bridge 00:01:47:93:74:54
```

**5** Create the default route.

See [Creating a default route on page 57](#).

**Deleting the IP on a bridge configuration****1** Verify the IP on a bridge interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-700
-----
```

**2** Delete the IP on a bridge interface.

```
zSH> interface delete 1/a/6/0/ip
Delete complete
```

This action automatically deletes the subscriber facing *ipobridge* *tls* bridge.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table	Data
	VLAN/SLAN	VLAN/SLAN					
tls		700	1/a/2/0/eth	ethemet2/bridge			UP

1 Bridge Interfaces displayed

**3** Delete the *tls* network facing bridge.

```
zSH> bridge delete ethernet2/bridge vlan 700
Bridge-path deleted successfully
ethernet2/bridge delete complete
```

**Configure IP on a bridge on a link aggregation bridge**

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24, and a logical port interface 6 on VLAN 200.

If you need to create a link aggregation group, see [Chapter 8, Link Aggregation Configuration](#) for link aggregation configuration rules and information.

**Creating IP on a bridge on a link aggregation bridge**

**1** Verify the link aggregation.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
a*  1  1-a-1-0          00:00:00:00:00:00    0x0    0x0    OOS    Active
    links          slot  port  subport          status
-----
    1-a-2-0          a     2     0                ACT
b   1  1-b-1-0          00:00:00:00:00:00    0x0    0x0    OOS    Active
    links          slot  port  subport          status
-----
    1-b-2-0          b     2     0                DSA
global linkagg group red type: red
```

**2** Create a *linkagg uplink* bridge. The uplink ports are the ports that are in the link aggregation.

```
zSH> bridge add 1-a-1-0/linkagg uplink vlan 200 tagged
Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1-200/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table  Data
-----
upl   Tagged 200  1/a/1/0/linkagg  linkagg-a-1-200/bridge  DWN S VLAN 200 default
1 Bridge Interfaces displayed
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

### 3 Enter **interface add** *interface/type* with the type as *ipobridge*.

This command creates the new IP interface as well as a new bridge. The bridge created will be a *downlink* tagged bridge.



**Note:** The logical port interface for IP on a bridge on the MXK *must* be *1-a-6-0/ipobridge* for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 200 192.168.8.21/24
Created ip-interface-record ipobridge-200/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

### 4 Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
```

### 5 Verify the *ipobridge*.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
upl   Tagged 200  1/a/1/0/linkagg  linkagg-a-1-200/bridge  DWN S VLAN 200 default
ipobdwn Tagged 200  1/a/6/0/ipobridge ipobridge-200/bridge  UP  S 00:01:47:93:74:54
                                           S 192.168.8.21
```

2 Bridge Interfaces displayed

A static IP and MAC address is created on the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge              Address
-----
200 linkagg-a-1-200/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
200 ipobridge-200/bridge  192.168.8.21
200 ipobridge-200/bridge  00:01:47:93:74:54
```

### 6 Create the default route.

See [Creating a default route on page 57](#).

## Deleting the IP on a bridge management interface

### 1 View the IP interface

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
-----
```

### 2 Delete the *ipobridge* interface.

```
zSH> interface delete 1/a/6/0/ip
Delete complete
```

This action automatically delete the *ipobridge downlink* bridge.

### 3 Delete the *linkagg* bridge.

```
zSH> bridge delete linkagg-a-1-200/bridge vlan 200
Bridge-path deleted successfully
linkagg-a-1-200/bridge delete complete
```

## Configure VoIP on IP on a bridge for EAPS

When configuring voice on an EAPS ring, you must use the IP address that you enter for the *ipobridge* interface.

### Configuring IP on a bridge for voice on an EAPS ring

- 1 Enter **interface add** *interface/type* with the type as *ipobridge*.

This command creates the new IP interface as well as a new *ipobridge* bridge. Entering the *tls* bridge type means that the *ipobridge* created will be a *tls* bridge.

```
zSH> interface add 1-a-6-0/ipobridge vlan 400 10.10.10.2/30 tls
Created ip-interface-record ipobridge-400/ip.
```

Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 10.10.10.2/30     00:01:47:93:74:54   ipobridge-400
-----
```

Verify the *ipobridge* that was created.

```
zSH> bridge show
          Orig
Type     VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
ipobtls          Tagged 400  1/a/6/0/ipobridge  ipobridge-400/bridge  UP  S 00:01:47:93:74:54
                                     S 10.10.10.2

1 Bridge Interfaces displayed
```

- 2 Create the default route for the *ipobridge* IP address.

```
zSH> route add default 10.10.10.1 1
```

### Create a default route

Regardless of which management IP interface is created, you must also create a default route for that interface.

### Creating a default route

Create the default route using the gateway 192.168.8.1 with a cost of 1 (one).

```
zSH> route add default 192.168.8.1 1
```

Verify the route:

```
zSH> route show
```

Destination Routing Table

Dest	Nexthop	Cost	Owner	Fallback
0.0.0.0/0	192.168.8.1	1	STATICLOW	
192.168.8.0/24	1/a/6/0/ip	1	LOCAL	

Use the **ping** command to verify connectivity to the default gateway:

```
zSH> ping 192.168.8.1
PING 192.168.8.1: 64 data bytes
!!!!
----192.168.8.1 PING Statistics----
5 packets transmitted, 5 packets received
round-trip (ms)  min/avg/max = 0/0/0
```

To stop the ping, press CTRL+C.

## Manage the MXK from ZMS

This section describes:

- [Configure the MXK to run ZMS in SNMPv3, page 59](#)
- [Mass provisioning from the CLI when running ZMS, page 60](#)

Before using Zhone Management System (ZMS), the Web UI or any remote management, a management interface must be configured for chassis access. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).

For ZMS refer to *NetHorizon User's Guide*, *ZMS Administrator's Guide*, and the *ZMS Installation Guide*. For OSS Gateway, refer to OSS Gateway documentation.

## Configure the MXK to run ZMS in SNMPv3

### Configuring the MXK to run ZMS in SNMPv3

In order to invoke SNMPv3 for ZMS, you must delete ZMS, update **system 0**, and rerunning ZMS.

- 1 Delete the device connected to ZMS that is running SNMPv2.
- 2 Update the **system 0** file on the MXK with the *snmpv3includingZMS* variable for the **snmpVersion** parameter by deleting ZMS.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {172.16.160.49_4_1392921484267}:
configsyncstatus: -----> {synccomplete}:
configsyncuser: -----> {zmsftp}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {172.16.160.49}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
```

```

primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {disdefpktrules}:
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}: snmpv3includingZMS
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

### 3 Reconnect the device to ZMS that is running SNMPv3.

In ZMS, open the region, select the correct region, then right-click **Add Device**.

From the **SNMP Version** drop-down menu in the **Add Device Configuration** dialog box, select **SNMP V3**.

## Mass provisioning from the CLI when running ZMS

In order to perform mass provisioning from the CLI when ZMS is running, you must disable partial config sync traps to ZMS from the device. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).



**Note:** For how to enable ZMS, refer to the *NetHorizon User's Guide*.

## CLI mass provisioning and ZMS

If you need to perform mass provisioning tasks with a script from the CLI when ZMS is managing the device, you must first disable ZMS in the **system 0** profile, complete the mass provisioning, enable ZMS again, and perform a config sync in ZMS.

### 1 Disable ZMS from managing the device, change the **zmsexists** parameter from *true* to *false*:

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:

```

```

zmsexists: -----> {true}: false
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> ** private **: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: -----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

## 2 Enable ZMS to manage the device, change the **zmsexists** parameter from *false* to *true*:

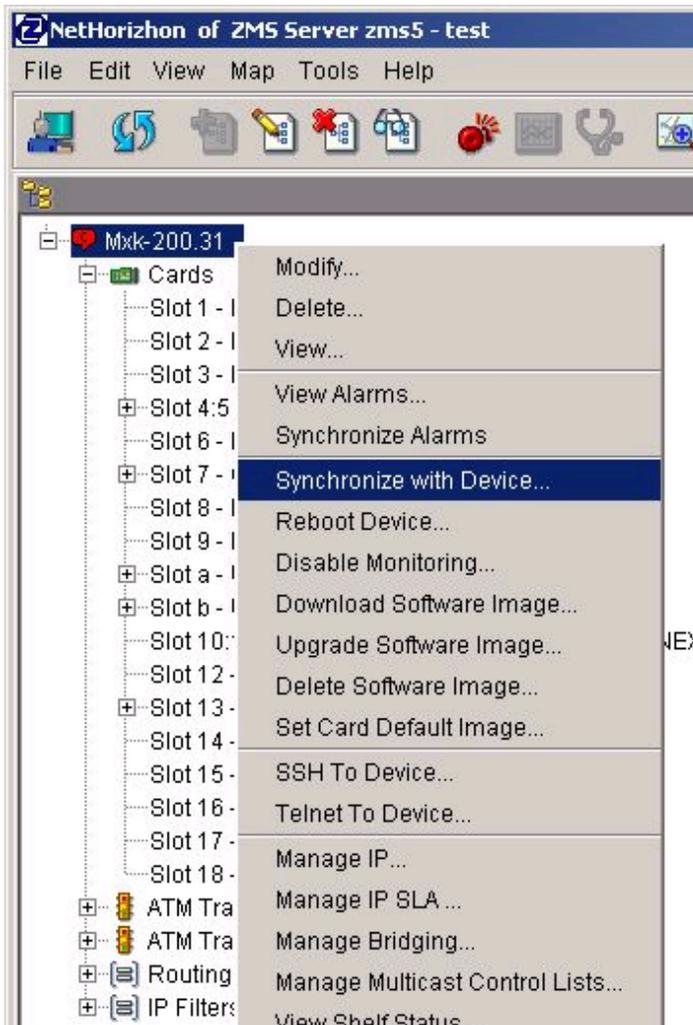
```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}: true
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:

```

```
configsyncaction: -----> {noaction}:
configsyncfilename: ----> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ----> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ----> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion:-----> {snmpv2}
persistentLogging: ----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

### 3 Perform a full config sync in ZMS.



**Note:** For details on using ZMS, refer to the *ZMS Administrator's Guide* and the *NetHorizon User's Guide*.

CLI configuration of a device being managed by the ZMS is disabled by default. Attempting to configure the device results in an error:

## Manage the MXK from the WebUI

This section describes:

- [Manage the MXK using Zhone Web User Interface, page 64](#)
- [Disable the Web UI, page 65](#)

Before using Zhone Management System (ZMS), the Web UI or any remote management, the management interface must be configured. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).



**Caution:** If you are using a public and not a private IP address for the Web UI, to protect your management system, Zhone recommends that the port access profile is configured for the Telnet port (port 23) and the management subnet is specified. See [Port access security on page 132](#) for more information on setting up port security.

The MXK enables Web-based configuration using the Zhone SLMS Web Interface Tool. The Zhone SLMS Web Interface Tool supports configuration and management of both line and uplink cards.

## Manage the MXK using Zhone Web User Interface

To manage the MXK using the Zhone Web User Interface (UI):

- Add an IP address to the interface to be used for management.

On the uplink cards, the interface on the 10/100 Ethernet port or GigE ports can be used. Ensure that the IP address is in the same subnet as the client devices and is reachable through Telnet. This example adds an IP interface for 172.24.94.103 to the 10/100 Ethernet port using VLAN 94.

```
zSH> interface add 1-a-1-0/eth vlan 94 172.24.94.103/24
Created ip-interface-record ethernet1-94/ip
```

- Configure a default route to the IP interface.

The default route enables connectivity to the IP interface.

```
zSH> route add default 94 172.24.94.103 metric 1
```

To launch the Zhone Web User Interface, in a browser URL address space on a PC with connectivity to the MALC, enter the IP address configured on the MXK.



**Note:** Ipv4 is required for all IP termination on the MXK, including management interfaces. IPv6 is not supported for IP termination on the MXK.

The Zhone Web User Interface launches and displays the Login window for the MXK.

Figure 4: Zhone Web User Interface Login Screen



Z H O N E

---

**MXK823**

---

The Multi-Access Line Concentrator, MXK, platform provides low-cost, high-density subscriber access concentration in the Zhone Single Line Multi-Service (SLMS) architecture. The MXK is designed as a high-bandwidth, non-blocking Ethernet based access platform that integrates high bandwidth access technologies (GPON and active Ethernet) of core/metro Ethernet networks with the breadth and flexibility of interfaces required at the edge of networks.

The MXK can be deployed in Central Office environments or controlled environmental vaults for remote terminal applications. The MXK is intended for restricted access locations only.

---

**Web Interface Login**

---

User Name

Password

**MXK 823**



On the Login page, enter the user name and password. The default user name is *admin* and the default password is *zhone*.



**Note:** Zhone recommends you change the user name and password to ones suitable to your network.

Click the desired menu to display the management options. For online help, click the Help icon  or product title in any window.

## Disable the Web UI

### Disabling the Web UI

Delete the *mxk823\_http.tar* or *mxk819\_http.tar* file from the *card1* directory to remove the software file associated with the Web UI. If you remove this file, you would have to reinstall the file in the *card1* directory to run the Web UI.

**1** Verify the current directory.

```
zSH> pwd
/card1
```

**2** Verify the *mxk823\_http.tar* or *mxk819\_http.tar* file in the *card1* directory.

```
zSH> dir
Listing Directory .:
-rwxrwxrwx 1 0      0          852028 Aug 22 11:51 mxup2tg8graw.bin
-rwxrwxrwx 1 0      0      13080567 Aug 22 11:51 mxup2tg8g.bin
-rwxrwxrwx 1 0      0      5310220 Sep  1 2011 mxlc48aadslbond.bin
-rwxrwxrwx 1 0      0      1100640 Jan 17 2011 malcmtac.bin
-rwxrwxrwx 1 0      0      1321250 Jan 17 2011 malcmtacenh.bin
-rwxrwxrwx 1 0      0      3788749 Jan 17 2011 mxlc48adsl2p.bin
-rwxrwxrwx 1 0      0      1322775 Jan 17 2011 tacitmring.bin
drwxrwxrwx 1 0      0         4096 Dec 21 2010 crash/
-rwxrwxrwx 1 0      0      4418987 Jan 17 2011 mxlcgp.bin
drwxrwxrwx 1 0      0         4096 Aug 22 13:35 datastor/
drwxrwxrwx 1 0      0         4096 Jan 17 2011 onreboot/
drwxrwxrwx 1 0      0         4096 Aug 22 13:34 log/
drwxrwxrwx 1 0      0         4096 Jul 27 2000 bulkstats/
drwxrwxrwx 1 0      0         4096 Jun  4 2010 pub/
-rwxrwxrwx 1 0      0      4257603 Sep  1 2011 mxlc24gshdslbond.bin
-rwxrwxrwx 1 0      0      5021611 Sep  1 2011 mxlc20ae.bin
-rwxrwxrwx 1 0      0      7341267 Aug 22 11:49 mxlc4gp.bin
drwxrwxrwx 1 0      0         4096 Jan 17 2011 me/
drwxrwxrwx 1 0      0         4096 Jan 17 2011 omci/
-rwxrwxrwx 1 0      0      405552 Jan 17 2011 mxlc20aerom.bin
-rwxrwxrwx 1 0      0      7341728 Aug 22 11:50 mxlc8gp.bin
-rwxrwxrwx 1 0      0      18428 Jan 17 2011 znid-gpon-2510-omci.txt
-rwxrwxrwx 1 0      0      9249280 Aug 22 11:48 mxk819_http.tar
-rwxrwxrwx 1 0      0      18428 Jan 17 2011 dumpme1
-rwxrwxrwx 1 0      0         748 Jan 17 2011 rsa.der
-rwxrwxrwx 1 0      0        1058 Jan 17 2011 rsakey.dat
drwxrwxrwx 1 0      0         4096 Jan 17 2011 newme/
drwxrwxrwx 1 0      0         4096 Jan 17 2011 1.16.2.123/
-rwxrwxrwx 1 0      0      9663488 Jan 17 2011 mxk823_http.tar
-rwxrwxrwx 1 0      0      5094732 Aug 22 11:48 mxlc20ae1s.bin
-rwxrwxrwx 1 0      0      7461652 Aug 22 11:49 mxlc24vds12.bin
-rwxrwxrwx 1 0      0      852028 Jan 17 2011 mxup8graw.bin
-rwxrwxrwx 1 0      0      5694994 Jan 17 2011 mxlc48badslbond.bin
147661088 bytes available
```

**3** Delete the *mxk823\_http.tar* file.

```
zSH> del mxk823_http.tar
```

The file is removed from the MXK. The file must be reinstalled in the *card1* directory to run the Web UI.

## MXK system administration

This section describes how to work with profiles in the MXK system, and many of the CLI commands that are useful when performing system administration tasks and includes a discussion of:

- [MXK system defaults, page 67](#)
- [User account administration, page 68](#)
- [View chassis and system information, page 73](#)
- [View runtime statistics for the MXK with the card stats command, page 79](#)
- [Monitor the system with log files, page 81](#)
- [Navigate the MXK file system, page 93](#)
- [MXK basic system administration commands, page 96](#)
- [Save and restore configurations, page 105](#)
- [SNTP, page 106](#)
- [MXK Simple Network Management Protocol \(SNMP\), page 108](#)

## MXK system defaults

This section describes the MXK system defaults, monitoring the MXK, and temporary logging sessions:

- [Defaults overview, page 67](#)
- [Monitoring the MXK through the serial craft port, page 68](#)
- [Enable/disable temporary logging sessions, page 68](#)

### Defaults overview

The MXK must have at least one uplink card installed before the MXK will boot properly. Along with the ability to display cards (both active and inactive) which are in the MXK, you can also see into the DOS file system which stores boot code, software images, and configurations. See [Navigate the MXK file system on page 93](#) for a description of commands which can be used to access the MXK file system.

Line cards (except the first uplink card in slot *a*) must be provisioned with a card-profile before they will boot up.

- Administrative user name is **admin**, password is **zhone**.
- A single record for the Ethernet interface on the uplink card in slot *a* exists. No other profiles to configure physical interfaces exist.

- The uplink card in slot *a* is enabled. You must enable all other cards including the uplink card in slot *b* in a **card-profile** before they will boot up.
- A default **system 0** profile exists with the following configuration:
  - Authentication traps are not enabled
  - ZMS communication is not configured
  - Alarm notification and output are enabled for all severity levels

## Monitoring the MXK through the serial craft port

The MXK can send messages to a console session, a log file, or to a syslog server and be configured to a number of system event levels — emergency, alert, critical, error, warning, notice, information, and debug.

## Enable/disable temporary logging sessions

By default, log messages are enabled on the serial craft port. Use the **log session** command and the **log serial** command to enable/disable logging:

The **log session** command enables/disables logging messages for that session only when connected to the device through a Telnet session. If the user logs out, the logging setting returns to the default. To enable/disable logging for the current Telnet session only enter:

```
zSH> log session on  
Logging enabled.
```

```
zSH> log session off  
Logging disabled.
```

This command setting does not persist across system reboots.

The **log serial** command enables/disables logging messages for the session on the serial craft port. This command can be used in both Telnet connections and serial port connections to turn on and off the serial craft port logs. To enable/disable logging for the serial craft port enter:

```
zSH> log serial on  
Serial port logging enabled.
```

```
zSH> log serial off  
Serial port logging disabled.
```

This command setting persists across system reboots.

## User account administration

MXK users have access to the CLI and are able to configure and administer the system.

- [Add users, page 69](#)
- [Create an SNMP v3 user from CLI, page 70](#)
- [Change default user passwords, page 70](#)
- [Delete users, page 70](#)
- [Delete the admin user account, page 71](#)
- [Reset passwords, page 71](#)

## Add users

Every administrative user on the system must have a user account. The account specifies their username and password, as well as their privilege level, which determines their access to commands.

Users with **admin** privileges have access to all the administrative commands. Users with **user** privileges have access to a very limited set of commands. The highest level of access is **useradmin**, which allows the creation of user accounts.




---

**Note:** When entering access level responses, enter **yes** completely or the CLI interprets the response as **no**.

---

To add a user, enter the following commands:

```
zSH> adduser
Please provide the following: [q]uit.
User Name: jjsmith
User Prompt [zSH>]:

Please select user access levels.
admin: -----> {no}: yes
zhonedebug: --> {no}:
voice: -----> {no}:
data: -----> {no}:
manuf: -----> {no}:
database: -----> {no}:
systems: -----> {no}:
tool: -----> {no}:
useradmin: ---> {no}: yes
.....
User name:(jjsmith) User prompt:(zSH>)
Access Levels:
(admin)(useradmin)
Save new account? [s]ave, [c]hange or [q]uit: s
User record saved.
TEMPORARY PASSWORD: hmj4mxFU
```

Commands with **zhonedebug** privilege levels are intended for use by Zhone development only.

Immediately after activating the user account, you should change the password something you can remember, as explained in the next section.

## Create an SNMP v3 user from CLI

### Creating an SNMP v3 user

- 1 Use the **adduser snmp *username*** command to create an SNMPv3 user. Select the Auth protocol and the Priv protocol, then enter a password if prompted.

For example:

```
zSH> adduser snmp test
Auth protocol (md5, sha, or none): md5
Enter auth password:
Confirm auth password:
Priv Protocol (des or none): des
Enter priv password:
Confirm priv password:
Enter access group (readwrite, readonly, encrypt, admin) : readwrite
```

- 2 Verify the user.

```
zSH> showuser snmp
```

userName	auth	priv	accessGroup
zmsUser	md5	des	readwrite
test	md5	des	readwrite

### Change default user passwords

When adding users, the system automatically assigns a temporary password to each user. Most users will want to change their password. The **changepass** command changes the password for the current logged in user. The following is an example of changing a password:

```
zSH> changepass
Current Password:
New Password:
Confirm New Password:
Password change successful.
```

### Delete users

To delete a user, enter the **deleteuser** command and specify the username:

```
zSH> deleteuser jsmith
OK to delete this account? [yes] or [no]: yes
User record deleted.
```

## Delete the admin user account

In addition to deleting regular user accounts, you can also delete the **admin** user account. This account is automatically created by the system and provides full access to the CLI.



**Note:** You cannot delete the **admin** account (or any other user account with **useradmin** privileges) if you are currently logged into it.

To delete the **admin** account:

```
zSH> deleteuser admin
```

If desired, you can recreate an account named **admin** after deleting it:

```
zSH> adduser admin
Please provide the following: [q]uit.
User Name: admin
User Prompt [zSH>]:

Please select user access levels.
admin: -----> {no}: yes
zhonedebug: --> {no}:
voice: -----> {no}: yes
data: -----> {no}: yes
manuf: -----> {no}: yes
database: -----> {no}: yes
systems: -----> {no}: yes
tool: -----> {no}: yes
useradmin: ---> {no}: yes
.....
User name: (admin)  User prompt: (zSH>)
Access Levels:
(admin) (voice) (data) (manuf) (database) (systems) (tools) (useradmin)
Save new account? [s]ave, [c]hange or [q]uit: s
User record saved.
TEMPORARY PASSWORD: hmj4mxFU
```

## Reset passwords

If a user forgets their password, an administrative user can reset the password and generate a new one using the **resetpass** command, as in the following example:

```
zSH> resetpass jsmith
Password:
```

## user command

The **user** command enables the command line feature to add, modify, show, or delete users and user settings.

**Syntax** `user add <user-name> [password string] [prompt string][admin] [debug] [voice] [data] [manuf] [dbase][systems][tools] [useradmin] [all]`

```
user modify <user-name> [password string] [prompt string][admin true|false] [debug true|false] [voice true|false][data true|false] [manuf true|false] [dbase true|false][systems true|false] [tools true|false] [useradmin true|false][all true|false]
```

changes user profile parameters

password option to set a new password

prompt option to set a new user prompt

other options set user access levels

"all" sets all access levels. It is processed before the other access level keywords, i.e. you can "manuf false all true".

That will set all access levels except manuf level access.

```
user delete <user-name>
```

deletes user account

```
user show [<user-name>]
```

displays user account information

displays all user accounts if no user-name entered

## Options add

Adds a new user profile with the specified settings.

### username

Name of the user.

### password password

Specifies the password assigned to this user.

### prompt

Specifies the system prompt to display for this user. If no password is entered, the system assigns a random password. Enclosing an argument in quotes allows the entry of special characters.

### access level

Specifies the access levels assigned to the user. The **all** option sets all access levels. Individual access levels can be specified by added the keyword true or false after an access level. For example, **manuf false all true** sets all access levels except manuf level access.

**Example 1**

```
zSH> user add steve password pass prompt "zSH >" admin voice systems dbase
User record saved.
.....
User name:(Steve)  User prompt:(zSH >)
Access Levels:
(admin) (voice) (system) (dbase)
```

**Example 2**

```
zSH> user modify joe password pass all false admin true
OK to modify this account? [yes] or [no]: yes
User record updated.
.....
User name:(newaccount2)  User prompt:(zSH>)
Access Levels:
(admin) (useradmin)
```

**Example 3**

```
zSH> user show
.....
User name:(admin)  User prompt:(zSH>)
Access Levels:
(admin) (voice) (data) (manuf) (database) (systems) (tool) (useradmin)
.....
User name:(steve)  User prompt:(zSH>)
Access Levels:
(admin) (voice) (systems) (dbase)
.....
User name:(joe)  User prompt:(test >)
Access Levels:
(admin)
.....
User name:(kathy)  User prompt:(test4 >)
Access Levels:
(admin) (zhonedebug) (voice) (data) (manuf) (database) (systems) (tool) (useradmin)

zSH> user show steve
.....
User name:(steve)  User prompt:(zSH>)
Access Levels:
(admin) (voice) (systems) (dbase)
```

**Example 4**

```
zSH> user delete kathy
OK to delete this account? [yes] or [no]: yes
Account kathy deleted
```

**View chassis and system information**

This section describes:

- [MXK 819 and 823 fan tray monitoring, page 74](#)
- [MXK 319 fan tray monitoring, page 75](#)
- [MXK built-in alarm input output, page 77](#)

## MXK 819 and 823 fan tray monitoring

The MXK supports monitoring the chassis/fan tray through the CLI.

The fan trays for the MXK 819 and MXK 823 support enhanced monitoring capabilities:

- individual fan rotation
- ambient air temperature
- three-point exhaust air temperature
- battery and return voltage measurement

To view overall status of the system, use the **shelfctrl monitor** command:

```
zSH> shelfctrl monitor
Shelf                               Status
-----
Uptime                               16 minutes
FPGA version                          0.5
Firmware version                      0.5
Uplink Supervisor CPLD version       1.3
Uplink Glue version                  0.2
16 MHz TDM clock                      Yes
Temperature Sensor                   Celsius (C)           Fahrenheit (F)
-----
Outlet sensor                         24                   75
Temperature reading                   normal
Fan Power Supplies & Alarm            Status
-----
Fan Power A                           normal
Fan Power B                           normal
Fan alarm                              ok
Power Supplies                        Status
-----
Battery A                             normal
Battery B                             normal
Device                                Status
-----
System                                Critical alarm set
Card a                                Critical alarm set
Alarm I/O Board
-----
Supported:                            No
Present:                              No
```

**System** and **Card a** will show **Critical alarm set** when an alarm has been triggered. Other parameters provide full descriptions such as **warning fans A, B, C, F are stopped** or **warning all fans are stopped** for the Fan alarm.

The Battery A and Battery B voltages are measured relative to battery return (+). The Battery return voltage measurement is relative to ground (i.e., the chassis).

Note that earlier versions of the MXK 819/MXK 823 fan tray do not support all the monitoring functionality shown here. Consult your Zhone sales person for more information. See [MXK built-in alarm input output on page 77](#) for a description of the Alarm I/O Board functionality.

## MXK 319 fan tray monitoring

The MXK 319 fan tray supports a subset of the monitoring features.

```
zSH> shelfctrl monitor
Shelf                                     Status
-----
Uptime                                   4 days, 3 hours, 29 minutes
FPGA version                             0.4
Firmware version                         0.0

Temperature Sensor                       Celsius (C)
Fahrenheit (F)

-----
Outlet sensor                            35                               95
Temperature reading                       normal

Fan Power Supplies & Alarm               Status
-----
Fan Power A                              normal
Fan Power B                              normal
Fan alarm                                 ok

Power Supplies                           Status
-----
Battery A                                normal
Battery B                                normal

Device                                    Status
-----
System
Card a
```

To verify whether the shelf is active:

```
zSH> shelfctrl show
```

```
Shelf Controller Address: 01:a:12
Shelf Registry Address: 01:a:1042
Lease ID: 0x02070000_00000033
State: active
Slot 1:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280841388
Slot 2:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 3:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 4:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 5:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280845212
Slot 6:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280910574
Slot 7:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280828845
Slot 8:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280837242
Slot 9:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 10:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280838736
Slot 11:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 12:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280828805
Slot 13:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot 14:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
Slot a:
  prevState: CONFIGURING currentState: RUNNING
  mode: FUNCTIONAL startTime: 1280828709
Slot b:
  prevState: NONE currentState: NONE
  mode: NONE startTime: 0
```

To view system statistics enter:

```

zSH> shelfctrl stats
Shelf Controller Message Statistics
-----
Directory services: 2
Clock: 275089
Lease: 1050
Heartbeat: 551264
Card status: 10
Info: 11
Card updates: 27

```

## MXK built-in alarm input output

Because the POTS line cards have both integrated ringing power and line test capabilities, the TAC card is no longer an essential component of installations except for the need for alarm inputs and reference clock inputs. To remove the need for alarm inputs, the new version MXK chassis has an alarm board with both input and output relays.

The *MXK Hardware Installation Guide* shows the location and description of the alarm input and output relays.

With the 2.3 release, the **shelfctrl monitor** command will display an **Alarm I/O Board** section at the bottom of the display. Note: the display has been truncated to show the new section (highlighted in bold).

```

zSH> shelfctrl monitorShelf
Status
-----
-----
Uptime                1 minute
FPGA version          0.5
Firmware version      0.6
Uplink Supervisor CPLD version 1.4
Uplink Glue version   0.2
16 MHz TDM clock      Yes
...
Device                Status
-----
-----
System                No alarms reported

Alarm I/O
Board-----
-----
Supported:           Yes
Present:            Yes
Alarm input:        Ai1  Ai2  Ai3  Ai4  Ai5  Ai6  Ai7  Ai8
Status (Energized/de-energized):  d   d   d   d   d   d   d
dNormalOpen/NormalClosed/NotSpec:  NS  NS  NS  NS  NS  NS  NS  NS
Alarm Active:      No   No   No
No   No   No

```

Older MXK chassis which do not have the Alarm I/O board running the 2.3 or newer software will show that the Alarm I/O board is not present (highlighted).

```
zSH> shelfctrl monitor
Shelf                               Status
-----
Uptime                               15 days, 23 hours, 34
minutes
FPGA version                         0.5
Firmware version                     0.5
Uplink Supervisor CPLD version      1.3
Uplink Glue version                  0.2
16 MHz TDM clock                     Yes
...
Device                               Status
-----
SystemNo alarms reported
Card aNo alarms reported

Alarm I/O Board
-----
Supported:                       No
Present:                         No
```

To support the Alarm I/O board, the correct uplink card and firmware needs to be present. For the 4x1G uplinks, the firmware is automatically upgraded when the software is upgraded to 2.3 or later.

The 8x1G and 2x10G+8x1G uplink cards do not upgrade automatically. Some of these uplinks with upgraded firmware are already in the field. To determine which uplink you have, use the **shelfctrl monitor** command:

- If the **shelfctrl monitor** display for **Alarm I/O Board** shows **Supported: Yes**, then **Present: Yes** then the alarm I/O board is present.
- If the **shelfctrl monitor** display for **Alarm I/O Board** shows **Supported: Yes**, the firmware is upgraded.
- If the **Alarm I/O Board** shows **Supported: No**, the uplink card does not support the alarm I/O board. Contact Zhone support.

**Adding a description to a chassis alarm**

The num2str-profile uses an index in the form:

```
/slot/282/alarm-contact
```

For the new MXK I/O alarm board, shelf must be 1, slot must be 0.

For example, the following example adds a description in the name field, and specifies normallyclosed in the normal-state field to the sixth alarm contact of the MXK i/o alarm board.

```
zSH> update num2str-profile 1/0/282/6
```

Please provide the following: [q]uit.

```
name: -----> {Relay 6}: cabinet open
```

```
normal-state: -> {notspecified}: normallyclosed
```

```
.....
```

Save changes? [s]ave, [c]hange or [q]uit: s

Record updated.

The **normal-state** field has three value options: **notspecified**, **normallyclosed**, **normallyopen**.

## View runtime statistics for the MXK with the card stats command

The **card stats** command displays runtime statistics for the MXK device.

```
zSH> card stats
----- cpu % utilization ----- memory (KB)----- Card
Memory      uptime
slot idle usage high  services framework  low  % Used Total  Peak  Avail
Status      ddd:hh:mm:ss   s/w version
=====
1      90    10    3     5      0      0    65.14  87227  56824  30410  1 -
OK      1:04:32:32  MXK 2.5.1.113
```

The **card stats all** command displays information for all the cards.

```
zSH> card stats all
----- cpu % utilization ----- memory (KB)----- Card
Memory      uptime
slot idle usage high  services framework  low  % Used Total  Peak  Avail
Status      ddd:hh:mm:ss   s/w version
=====
2      97     3     1     0      0      3    34.71 100770  34987  65793  1 -
OK      6:22:11:51  MXK 2.5.1.113
3      99     1     0     0      0      0    13.85 121685  16854 104832  1 -
OK      6:22:11:57  MXK 2.5.1.113
4      92     8     4     2      0      0    40.05 104662  41923  62749  1 -
OK      6:22:11:10  MXK 2.5.1.113
5      92     8     5     2      0      0    42.54 104596  44507  60100  1 -
OK      6:22:10:17  MXK 2.5.1.113
6      92     8     6     1      0      2    34.01 109718  37320  72407  1 -
OK      6:22:12:29  MXK 2.5.1.113
10     85    15     0    14      0      0    35.33 107438  38064  69476  1 -
OK      6:22:10:25  MXK 2.5.1.113
```

```

a* 85 15 3 11 0 0 38.52 210359 81059 129334 1 -
OK 6:22:13:47 MXK 2.5.1.113
    
```

**Table 3: card stats command fields**

Section	Field
<b>CPU % utilization</b>	<b>slot</b> Textual description of the unit/card or access device type.
	<b>idle</b> Percentage of time the CPU has spent executing tasks with priority of 200 or less. Tasks with priority of 200 or less (the higher the number, the lower the priority) are considered idle tasks.
	<b>usage</b> Percentage of time the CPU has spent executing tasks with priority of 199 or higher
	<b>high</b> High priority tasks are primarily related to packet processing and critical system monitoring. Percentage of time the CPU has spent executing tasks with priority of 001 to 099. High priority tasks are primarily related to packet processing and critical system monitoring.
	<b>services</b> Services are primarily line monitoring tasks for line state and alarms. Percentage of time the CPU has spent executing tasks with priority of 100 to 179. Services tasks are primarily line monitoring tasks for line state and alarms.
	<b>framework</b> Framework tasks are primarily database and network management system related activities such as config synch and backup. Percentage of time the CPU has spent executing tasks with priority of 180 to 199. Framework tasks are primarily database and network management system related activities such as config synch and backup.
	<b>low</b> Percentage of time the CPU has spent executing tasks with priority of 200 to 250
	<b>memory (KB)</b>
<b>Used</b> Percentage of time the CPU has spent executing tasks with priority of 199 or higher.	
<b>Total</b> The amount of physical memory contained by the device/card.	

Table 3: card stats command fields (Continued)

Section	Field
	<p><b>Peak</b></p> <p>The maximum physical memory that has been allocated at any time by the device/card.</p>
	<p><b>Avail</b></p> <p>The amount of physical memory that is unallocated and not in use by the device/card.</p>
<b>Card Memory Status</b>	<p>Memory status of the card sent with memory trap. A trap is sent when each condition occurs.</p> <p>1 - ramMemOK      less then 90% of ram is used</p> <p>2 - ramMemLow     more then 90% of ram is used</p> <p>3 - flashMemOK    enough flash for maximum database</p> <p>4- flashMemLow    not enough flash for maximum database</p> <p>5 - flashMemOut    no more flash memory, data no longer persistent</p>
<b>uptime ddd:hh:mm:ss</b>	Uptime is calculated as sysUpTime - ifLastChange (assuming the device/card is running).
<b>s/w version</b>	Software version.

## Monitor the system with log files

This section provides the following information on how logs work on the MXK

- [Overview, page 82](#)
- [Default log store level, page 82](#)
- [User login notification, page 82](#)
- [Enable/disable temporary logging sessions, page 68](#)
- [Log message format, page 83](#)
- [Modify logging levels, page 84](#)
- [Non-persistent log messages, page 86](#)
- [Persistent log messages, page 88](#)
- [Example log messages, page 88](#)
- [Log filter command, page 88](#)
- [Send messages to a syslog server, page 89](#)
- [Specify different log formats for system and syslog messages, page 91](#)

## Overview

Logging enables administrators to monitor system events by generating system messages. It sends these messages to:

- A temporary management session (either on the serial craft port or over a Telnet session)
- Log modules to create permanent log files
- A syslog server (optional)

The type of information sent in these messages can be configured using the **log** command. By default, the system sends the same type of information to all log message destinations. If you want to send different types of messages to the syslog daemon, use the **syslog** command.

## Default log store level

The default log store level is now set to emergency so by default the **log display** command displays only emergency level messages. Use the **log cache** command to display all messages that have been logged to console.

Use the **cd log** and **dir** commands to view the log file history. The log files in this directory record console activity on the MXK for the running image, and preserve a copy of the last two reboots. The files *consolelog1.txt* and *consolelog2.txt* hold 10000 lines of console output each. Once the file reaches 10000 lines, the filename is changed to *.old* and a new *.txt* file is used. After a reboot, the *.txt* files are also saved as *.old* files. Use the **consolelog display <filename>** command to view the contents for a consolelog file. These files are used for troubleshooting and system activity monitoring.

## User login notification

Notifications of user login are sent to the console log.

```
zSH> MAR 11 17:28:20: alert : 1/a/1031: clitask1: User admin@172.16.48.232 logged in on slot a
```

## Enable/disable logging

By default, log messages are enabled on the serial craft port. Use the **log session** command and the **log serial** command to enable/disable logging:

The **log session** command enables/disables logging messages for that session only. If the user logs out, the logging setting returns to the default. To enable logging for the current session only:

```
zSH> log session on
Logging enabled.
```

To disable logging for the session:

```
zSH> log session off
```

Logging disabled.

The **log serial** command enables/disables logging messages for all sessions on the serial craft port. This setting persists across system reboots. To enable/disable logging for the serial craft port:

```
zSH> log serial on
Serial port logging enabled.
```

To disable logging for the serial port:

```
zSH> log serial off
Serial port logging disabled.
```

## Log message format

Log messages contain the following information:

**Table 4: Default log message fields**

Option	Description
Date	Date stamp of log message. Enabled by default.
Time	Time stamp of log message. Enabled by default.
Ticks	Current tick count. When the <b>tick</b> option is used, the date and time fields are not displayed.
Level	Logging level of the message. Enabled by default.
Address	The shelf and slot and application identifier causing the alarm.
Logtest	Log handle.
Taskname	Name of task that generated the log message. This is generally useful only for Zhone development engineers. Enabled by default.
Function	Function that generated the log message.
Line	Line in code that generated the log message. This is generally useful only for Zhone support staff.
Port	Port related to the log message.
Category	Category of the log message.
System	System related to the log message.
All	Controls all log message options.
Default	Controls the default log message options.
Message text	A description of the error that caused the alarm.

To change the information displayed in the log messages, use the **log option** command. First, display the available options:

```
zSH> log option
```

```
Usage: log option < time      | 1 > < on | off >
      < date      | 2 > < on | off >
      < level     | 3 > < on | off >
      < taskname  | 4 > < on | off >
      < taskid   | 5 > < on | off >
      < file     | 6 > < on | off >
      < function  | 7 > < on | off >
      < line     | 8 > < on | off >
      < port     | 9 > < on | off >
      < category  | 10 > < on | off >
      < system   | 11 > < on | off >
      < ticks    | 12 > < on | off >
      < stack    | 13 > < on | off >
      < globalticks | 14 > < on | off >
      < all      | 14 > < on | off >
      < default  | 15 > < on | off >
```

options 'time' & 'date' supercede option 'ticks'

time: date: level: address: log: port: category: system: (0x707)

Then, turn the option **on** or **off**. For example, the following command will turn the task ID on or off in log messages:

```
zSH> log option taskid on
```

time: date: level: address: log: taskid: port: category: system: (0x717)

```
zSH> log option taskid off
```

time: date: level: address: log: port: category: system: (0x707)

The following commands will turn on or off the tick count display in log messages:

```
zSH> log option ticks on
```

time: date: level: address: log: port: category: system: ticks: (0xf07)

```
zSH> log option ticks off
```

time: date: level: address: log: port: category: system: (0x707)

The following command will turn all options on in log messages:

```
zSH> log option all on
```

time: date: level: address: log: taskname: taskid: file: function: line: port: category: system: ticks: stack: globalticks: (0x3fff)

## Modify logging levels

To modify logging, use the **log** command. To modify syslog messages, use the **syslog** command.



**Caution:** Changing the log level may generate enough output to disrupt service.

---

To display the current levels for all logging modules, use the **log show** command:

```
zSH> log show
MODULE                                     LEVEL      STATUS
adslhdlr                                   error      enabled
adslprof                                   error      enabled
alarm_mgr                                  error      enabled
assert                                     error      enabled
atm_cc_mib_hdlr                            error      enabled
atmmgragnt                                 error      enabled
bds                                         error      enabled
bds_client                                 error      enabled
bridge                                     error      enabled
bridgemib                                  error      enabled
bridgerp                                   error      enabled
bulkstats                                  error      enabled
bulkstatshdlr                              error      enabled
cam                                         error      enabled
card                                        error      enabled
card_resource                              error      enabled
carddeletehdlr                             info       enabled
cardred                                    error      enabled
cardsvchdlr                                error      enabled
ccrp                                        error      enabled
ccrr                                        error      enabled
cesmibhdlr                                 error      enabled
cli                                         error      enabled
clkmgr                                     warning    enabled
.....
```

Logging levels determine the number of messages that are displayed on the console. The higher the log level, the more messages are displayed. The MXK supports the following log levels:

- 1: emergency
- 2: alert
- 3: critical
- 4: error
- 5: warning
- 6: notice
- 7: information
- 8: debug

To change the log level, use the **log module level** command. For example, the following command changes the card module logging level to emergency:



**Caution:** Changing the log level may generate enough output to disrupt service.

```
zSH> log level card emergency
Module: card at level: emergency
```

To enable or disable log levels for a module, use the `log enable` or `log disable` commands. For example:

```
zSH> log disable card
Module: card is now disabled
```

## Non-persistent log messages

The `log cache` command displays the non-persistent log cache messages:

```
zSH> log cache
[1]: MAY 19 14:28:31: alert   : 1/a/1025: alarm_mgr: 01: a:06 Critical ETHERNET Down -
Ethernet line down
[2]: MAY 19 14:30:19: alert   : 1/13/1025: alarm_mgr: 01:13:01 Major ETHERNET Up -
Ethernet line up
[3]: MAY 19 14:32:12: alert   : 1/13/1025: alarm_mgr: 01:13:01 Major ETHERNET Down -
Ethernet line down
[4]: MAY 19 14:32:26: alert   : 1/13/1025: alarm_mgr: 01:13:02 Major ETHERNET Up -
Ethernet line up
[5]: MAY 19 14:33:27: alert   : 1/13/1025: alarm_mgr: 01:13:02 Major ETHERNET Down -
Ethernet line down
[6]: MAY 19 14:36:23: alert   : 1/4/1025: alarm_mgr: 01: 4:01:01 Minor ONU Down
Line 1/4/1/1/gpononu CAUSE: inactive
[7]: MAY 19 14:36:32: alert   : 1/4/1025: alarm_mgr: 01: 4:01:01 Minor ONU Up
Line 1/4/1/1/gpononu CAUSE: active
[8]: MAY 19 14:36:53: critical: 1/a/1035: rebootserver:
* * * * Slot Reboot : type = 2, shelf = 1, slot = 4
[9]: JAN 01 00:00:11: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c0
[10]: JAN 01 00:00:11: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c1
[11]: JAN 01 00:00:12: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c2
[12]: MAY 19 14:40:32: notice  : 1/a/12   : shelfctrl: Card in slot 4 changed state to
RUNNING.
[14]: MAY 19 14:40:32: alert   : 1/4/1025: alarm_mgr: 01: 4:02 Critical OLT Up

Line 1/4/2/0/gponolt CAUSE: active
```

The `log cache max length` command sets the maximum number of log messages to store. The maximum log cache size is 2147483647, depending in the amount of memory available.

### log cache max length

To change the current configured log cache size:

```
zSH> log cache max 200
```

Maximum number of log messages that can be saved: 200

The **log cache grep pattern** command searches through the log cache for the specified regular expression.

**log cache grep pattern**

The following example searches through the log cache for the string "Critical":

```
zSH> log cache grep Critical
```

Searching for: "Critical"

```
[1]: AUG 02 22:37:19: alert : 1/a/1025: alarm_mgr: 01: a:01 Critical ETHERNET Up -
Ethernet line up
```

```
[2]: AUG 02 22:37:34: alert : 1/a/1025: alarm_mgr: 01: a:02 Critical ETHERNET Down -
Ethernet uplink down
```

```
[3]: AUG 02 22:37:34: alert : 1/a/1025: alarm_mgr: 01: a:03 Critical ETHERNET Down -
Ethernet line down
```

The **log cache clear** command clears the log cache.

**log cache clear**

The **log cache size** command sets the maximum amount of memory for the log cache. Without options, displays the current log size.

```
zSH> log cache size
```

Number of log messages in the cache: 20

Total bytes used by the cache: 2052

The **log cache help** command displays the help information for the **log cache** command:

```
zSH> log cache help
```

```
Usage: log cache < max > < length >
        < grep > < pattern >
        < clear >
        < size >
        < help >
```

With no arguments the 'log cache' command prints out all the log messages currently in the cache.

The 'max' command is used to view/set the maximum number of log messages that can be cached at one time. If the cache is full then the oldest log is discarded and the new log is inserted. If no value is given then the current setting is displayed

The 'size' command is used to display the amount of memory currently being used by the log cache.

The 'clear' command is used to erase the log cache.

The 'grep' command is used for searching the log cache for a specific pattern. Extended regular expressions are supported.

## Persistent log messages

Use the **log cache** command to view the persistent logs which only stores emergency level logs. For example:

```
zSH> log display
AUG 07 19:01:17: emergency: 1/a/12 : shelfctrl: Critical alarm set!
AUG 07 21:25:36: emergency: 1/a/12 : shelfctrl: Critical alarm set!
SEP 21 17:44:22: emergency: 1/a/12 : shelfctrl: Critical alarm set!
NOV 19 18:58:18: emergency: 1/a/12 : shelfctrl: Critical alarm set!
NOV 22 03:30:37: emergency: 1/a/12 : shelfctrl: Critical alarm set!
DEC 06 18:23:37: emergency: 1/a/12 : shelfctrl: Critical alarm set!
FEB 13 21:00:45: emergency: 1/a/12 : shelfctrl: Critical alarm set!
MAR 04 19:07:32: emergency: 1/a/12 : shelfctrl: Critical alarm set!
```

## Example log messages

This section provides examples of how to interpret log messages.

The following message appears when a card in the MXK chassis comes up or goes down.

The most important parts of the message are the date and time the event occurred, the shelf/slot of the event, and the message text. The remainder of the information is only useful for Zhone development engineers.

For example:

```
MAR 11 17:46:20: alert : 1/6/1025: alarm_mgr: 01: 6:01
Minor ETHERNET Down - Ethernet line down
```

```
MAR 11 17:46:21: alert : 1/6/1025: alarm_mgr: 01: 6:01
Minor ETHERNET Up - Ethernet line up
```

```
MAR 11 17:48:30: alert : 1/5/1025: alarm_mgr: 01: 5:03
Critical OLT Up
Line 1/5/3/0/gponolt CAUSE: active
```

## Log filter command

The **log filter** command is available as part of the log command functionality. This command enables users to show, set and delete log filters. Log filters limit the scope of log messages to a specific entity for troubleshooting and diagnostics. When a log filter is set, the filter is assigned an index number and only messages relate the specified entity are displayed. Filters can be set for an specific ifindex, slot/port, VCL, or subscriber.

## log filter

Restrict the display of log messages to only the log messages for a specified entity.

```
Syntax log filter show | set (ifindex|port slotport|vcl ifindex
vpi vci|subscriber endpoint) | delete
```

```
zSH> log filter set ifindex 12
New filter saved.
```

```
zSH> log filter set port 5 24
New filter saved.
```

```
zSH> log filter set subscriber 22
New filter saved.
```

```
zSH> log filter show
Index   Type           Filter Parameters
-----
1       Port           slot=1, port=1
2       Port           slot=1, port=4
3       IfIndex        IfIndex=12
4       Port           slot=5, port=24
6       IfIndex        IfIndex=100
7       IfIndex        IfIndex=104
8       IfIndex        IfIndex=109
9       IfIndex        IfIndex=103
10      IfIndex        IfIndex=107
```

```
zSH> log filter delete 10
Log filter 10 deleted
```

## Send messages to a syslog server

[Table 5](#) describes the parameters in the **syslog-destination** profile you can modify to send messages to a syslog server.

**Table 5: syslog-destination profile parameters**

Parameter	Description
<b>address</b>	The IP address of the machine hosting the syslog server. Default: <b>0.0.0.0</b>
<b>port</b>	The UDP port to which the syslog messages will be sent. Default: <b>514</b>

**Table 5: syslog-destination profile parameters (Continued)**

Parameter	Description
<b>facility</b>	<p>The syslog facility to which the syslog messages will be sent.</p> <p>Values:</p> <ul style="list-style-type: none"> <li><b>local0</b></li> <li><b>local1</b></li> <li><b>local2</b></li> <li><b>local3</b></li> <li><b>local4</b></li> <li><b>local5</b></li> <li><b>local6</b></li> <li><b>local7</b></li> <li><b>no-map</b></li> </ul> <p>Default: <b>local0</b></p>
<b>severity</b>	<p>The severity level used to filter messages being set to the syslog server.</p> <p>Values:</p> <ul style="list-style-type: none"> <li><b>emergency</b></li> <li><b>alert</b></li> <li><b>critical</b></li> <li><b>error</b></li> <li><b>warning</b></li> <li><b>notice</b></li> <li><b>info</b></li> <li><b>debug</b></li> </ul> <p>Default: <b>debug</b></p>

```

zSH> new syslog-destination 1
Please provide the following: [q]uit.
address: --> {0.0.0.0}: 192.200.42.5 IP address of the syslog server
port: -----> {514}: leave at default
facility: -> {local0}:
severity: -> {debug}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
    
```

## Specify different log formats for system and syslog messages

Table 6 describes the **log-module** profile that supports the configuration of persistent log messages, syslog messages, and persistent storage levels by module. Modify this profile when you need to send different messages to admin sessions, the persistent logs, and the syslog server.

**Table 6: log-module profile parameters**

Parameter	Description
<b>name</b>	The name of the module whose logging is controlled by this profile. Default: <b>logtest</b>
<b>display</b>	Controls the display of messages on the system. Messages logged at this level and above will be displayed. Values: <b>emergency</b> <b>alert</b> <b>critical</b> <b>error</b> <b>warning</b> <b>notice</b> <b>info</b> <b>debug</b> Default: <b>error</b>

**Table 6: log-module profile parameters (Continued)**

Parameter	Description
<b>syslog</b>	<p>Controls the format of messages sent to the syslog server described in the <b>syslog-destination</b> profile. This field is similar to the display field, except for the <b>trackdisplay</b> value.</p> <p>Values:</p> <ul style="list-style-type: none"> <li><b>emergency</b></li> <li><b>alert</b></li> <li><b>critical</b></li> <li><b>error</b></li> <li><b>warning</b></li> <li><b>notice</b></li> <li><b>info</b></li> <li><b>debug</b></li> </ul> <p><b>trackdisplay</b> Messages logged at, and above, the level set in the <b>display</b> parameter will also be recorded in the syslog server.</p> <p>Default: <b>trackdisplay</b></p>
<b>store</b>	<p>Controls the persistent storage of messages. This field is similar to the display field, except for the <b>trackdisplay</b> value.</p> <p>Values:</p> <ul style="list-style-type: none"> <li><b>emergency</b></li> <li><b>alert</b></li> <li><b>critical</b></li> <li><b>error</b></li> <li><b>warning</b></li> <li><b>notice</b></li> <li><b>info</b></li> <li><b>debug</b></li> </ul> <p><b>trackdisplay</b> Messages logged at, and above, the level set in the <b>display</b> parameter will also be recorded in the syslog server.</p> <p>Default: <b>trackdisplay</b></p>

```

zSH> new log-module 1
Please provide the following: [q]uit.
name: ----> {logtest}: test1
display: -> {error}: warning
syslog: --> {trackdisplay}:
store: ---> {trackdisplay}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
    
```

In this case, the **log-module 1** will display to the screen, all messages at and above *warning*. The variable *trackdisplay* means that the same messages as defined in display are also sent to the syslog and storage. If different level of

messages are needed for the different destinations, the variables for **display**, **syslog**, and **store** can be set at different levels.

## Navigate the MXK file system

This section describes the MXK file system and includes:

- [Access the MXK file system, page 93](#)
- [Download software files, page 94](#)

## Access the MXK file system

Use the following commands to access the MXK file system:

- **cd** Changes directory.
- **dir** Lists the contents of the directory.
- **pwd** Displays the current working directory.
- **image** Verifies software images and downloads software images on the flash to system memory.

The uplink card flash memory contains DOS file system that stores the system boot code, software images, and the configuration. During system startup, the software images on the flash are decompressed and loaded into memory.

Use the **cd**, **dir**, and **pwd** commands to list the contents of the file system, as in the following example:

Change directory.

```
zSH> cd /card1
```

Print the working directory.

```
zSH> pwd
/card1
```

List the directories in the current directory.

```
zSH> dir
Listing Directory .:
-rwxrwxrwx 1 0      0      852028 Aug 22 11:51 mxup2tg8graw.bin
-rwxrwxrwx 1 0      0     13080567 Aug 22 11:51 mxup2tg8g.bin
-rwxrwxrwx 1 0      0     5310220 Sep  1 2011 mxlc48aadslbond.bin
-rwxrwxrwx 1 0      0     1100640 Jan 17 2011 malcmtac.bin
-rwxrwxrwx 1 0      0     1321250 Jan 17 2011 malcmtacenh.bin
-rwxrwxrwx 1 0      0     3788749 Jan 17 2011 mxlc48adsl2p.bin
-rwxrwxrwx 1 0      0     1322775 Jan 17 2011 tacitmring.bin
drwxrwxrwx 1 0      0      4096 Dec 21 2010 crash/
-rwxrwxrwx 1 0      0     4418987 Jan 17 2011 mxlcgp.bin
drwxrwxrwx 1 0      0      4096 Aug 22 13:35 datastor/
drwxrwxrwx 1 0      0      4096 Jan 17 2011 onreboot/
drwxrwxrwx 1 0      0      4096 Aug 22 13:34 log/
```

```

drwxrwxrwx 1 0      0          4096 Jul 27 2000 bulkstats/
drwxrwxrwx 1 0      0          4096 Jun  4 2010 pub/
-rwxrwxrwx 1 0      0        4257603 Sep  1 2011 mxlc24gshdslbond.bin
-rwxrwxrwx 1 0      0        5021611 Sep  1 2011 mxlc20ae.bin
-rwxrwxrwx 1 0      0        7341267 Aug 22 11:49 mxlc4gp.bin
drwxrwxrwx 1 0      0          4096 Jan 17 2011 me/
drwxrwxrwx 1 0      0          4096 Jan 17 2011 omci/
-rwxrwxrwx 1 0      0        405552 Jan 17 2011 mxlc20aerom.bin
-rwxrwxrwx 1 0      0        7341728 Aug 22 11:50 mxlc8gp.bin
-rwxrwxrwx 1 0      0         18428 Jan 17 2011 znid-gpon-2510-omci.txt
-rwxrwxrwx 1 0      0        9249280 Aug 22 11:48 mxk819_http.tar
-rwxrwxrwx 1 0      0         18428 Jan 17 2011 dumpme1
-rwxrwxrwx 1 0      0          748 Jan 17 2011 rsa.der
-rwxrwxrwx 1 0      0         1058 Jan 17 2011 rsakey.dat
drwxrwxrwx 1 0      0          4096 Jan 17 2011 newme/
drwxrwxrwx 1 0      0          4096 Jan 17 2011 1.16.2.123/
-rwxrwxrwx 1 0      0        9663488 Jan 17 2011 mxk823_http.tar
-rwxrwxrwx 1 0      0        5094732 Aug 22 11:48 mxlc20ae1s.bin
-rwxrwxrwx 1 0      0        7461652 Aug 22 11:49 mxlc24vdsl2.bin
-rwxrwxrwx 1 0      0         852028 Jan 17 2011 mxup8graw.bin
-rwxrwxrwx 1 0      0        5694994 Jan 17 2011 mxlc48badslbond.bin
147661088 bytes available

```

## Download software files

The MXK contains a TFTP client that enables you to download files from a network to the flash card file system using the **image** command. A software image for the uplink card and each type of line card must be downloaded.

The **image** command uses the following syntax:

```
image download tftphost imagefilename
```

The following example downloads the software image for the uplink card (*mxkup2tg8graw.bin*) from host 192.168.8.21 to the root directory of the first flash card:

```
image download 192.168.8.21 mxup2tg8graw.bin
```

## Downloading software files

Download software files from the TFTP server to the MXK when you need to upgrade the system software:

- 1 Create the *onreboot* directory if one does not already exist and back up the current configuration file to the a file named *restore*, then **cd** back to the root directory.

```

zSH> mkdir onreboot
zSH> cd onreboot
zSH> dump file restore
zSH> cd ..

```

The *restore* file is used to restore the system configuration or revert to a previous release, if desired. See [Step 5](#).

- 2 Copy the new system boot image software to the flash memory using the **image download** command.

```
zSH> image download 192.168.8.21 mxup2tg8g.bin
```

where 192.168.8.21 is the TFTP server, and *mxup2tg8g.bin* is the name of the software image.



**Caution:** Be sure to download the correct software for the system.

---

- 3 Initialize the flash card's boot partition with the new image on both the primary and standby uplink card (if present).

For a single uplink card enter:

```
zSH> image flash mxup2tg8g.bin 1 1
```

For redundant uplink cards enter:

```
zSH> image flash mxup2tg8g.bin 1 all
```

- 4 The **image** command can also verify image files on the flash card. It reads the contents of the file, verifies the file header, and verifies the file checksum. For example:

```
zSH> image verify mxup2tg8g.bin
File: mxup2tg8graw.bin
Size: 688320 bytes
Header Version: 1
Card Type: MX TWO TENGIGE EIGHT GIGE
Load Type: binary
Load Address: 0x00010000
Checksum: 0x2f66bb70
Image verify successful
```

The command reports any errors it finds in the file. Note that files are also verified as part of the download process.

- 5 Reset the system and restore the system configuration with the **systemreboot** command:

```
zSH> systemreboot
A restore file (/card1/onreboot/restore) is present.
A system reboot will result in a database restore.
Continue? (yes or no) [no]: yes
Do you want to reboot the system? (yes or no) [no] yes
Do you want to exit from this request? (yes or no) [yes] no
Are you sure? (yes or no) [no] yes
```

As shown above, when the *restore* file is present, the system displays

```
A restore file (/card1/onreboot/restore) is present.
```

and uses that file to restore the saved configuration to the MXK system.

After upgrading the software, the system automatically upgrades the software database to the new level.

## MXK basic system administration commands

### Commands: new, list, show, get, update, delete

This section describes these commands:

- [new command, page 96](#)
- [list command, page 96](#)
- [show command, page 99](#)
- [get command, page 101](#)
- [update command, page 102](#)
- [delete command, page 102](#)

#### ***new command***

The **new** command can create new GPON traffic profiles.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}:
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

#### ***list command***

The **list** command displays all the profiles available on the MXK (partial list shown):

```
zSH> list
adsl-co-profile: shelf/slot/port
adsl-cpe-profile: shelf/slot/port
adsl-profile: shelf/slot/port
alarm-config: ifIndex
analog-fxo-cfg-profile: ifIndex
analog-fxs-cfg-profile: ifIndex
analog-if-cfg-profile: ifIndex
atm-cc: atmVcCrossConnectIndex
atm-if: ifIndex
atm-if-stats: ifIndex
atm-traf-descr: index
atm-traf-descr-stats: index
atm-vcl: ifIndex/vpi/vci
```

```

atm-vcl-param: index
atm-vcl-stats: ifIndex/vpi/vci
atm-vpi: ifIndex/vpi
atm-vpl: ifIndex/vpi
bridge-interface-record: ifIndex
bulk-statistic: index

```

The **list gpon-traffic-profile** command lists all GPON traffic profiles on the system.

```

zSH> list gpon-traffic-profile
gpon-traffic-profile 1
gpon-traffic-profile 2
gpon-traffic-profile 3
3 entries found.

```

The **list system** command displays the list of system profiles.

```

zSH> show system
syscontact:-----> {260}
sysname:-----> {260}
syslocation:-----> {260}
enableauthtraps:-----> enabled disabled
setserialno:-----> {0 - 2147483647}
zmsexists:-----> true false
zmsconnectionstatus:--> active inactive
zmsipaddress:-----> {0 - 0}
configsyncexists:-----> true false
configsyncoverflow:---> true false
configsyncpriority:---> none low medium high
configsyncaction:-----> noaction createlist createfulllist
configsyncfilename:---> {68}
configsyncstatus:-----> synccomplete syncpending syncerror syncinitializing
configsyncuser:-----> {36}
configsyncpasswd:-----> {36}
numshelves:-----> {0 - 0}
shelvesarray:-----> {36}
numcards:-----> {0 - 0}
ipaddress:-----> {0 - 0}
alternateipaddress:---> {0 - 0}
countryregion:-----> argentina australia belgium china costarica finland
france germany hongkong italy japan korea mexico netherlands newzealand
singapore spain sweden switzerland uk us afghanistan albania algeria
americansamoa andorra angola anguilla antarctica antiguabarbuda armenia aruba
austria azerbaijan bahamas bahrain bangladesh barbados belarus belize benin
bermuda bhutan bolivia bosniaherzegovina botswana bouvetisland brazil
britishindianoceanterritory bruneidarussalam bulgaria burkinafaso burundi
cambodia cameroon canada capeverde caymanislands centralafricanrepublic chad
chile christmasisland cocosislands colombia comoros congo cookislands
cotedivoire croatia cuba cyprus czechrepublic denmark djibouti dominica
dominicanrepublic easttimor ecuador egypt elsalvador equatorialguinea eritrea
estonia ethiopia falklandislands faroeislands fiji frenchguiana frenchpolynesia
frenchsouthernterritories gabon gambia georgia ghana gibraltar greece greenland
grenada guadeloupe guam guatemala guinea guineabissau guyana haiti

```

```

heardislandmcdonaldislands holysee honduras hungary iceland india indonesia
iran iraq ireland israel jamaica jordan kazakistan kenya kiribati northkorea
kuwait kyrgyzstan lao latvia lebanon lesotho liberia libyanarabjamahiriya
liechtenstein lithuania luxembourg macau macedonia madagascar malawi malaysia
maldives mali malta marshallislands martinique mauritania mauritius mayotte
micronesia moldova monaco mongolia montserrat morocco mozambique myanmar
namibia nauru nepal netherlandsantilles newcaledonia nicaragua niger nigeria
niue norfolkisland northernmarianaislands norway oman pakistan palau
palestinianterritory panama papuanewguinea paraguay peru philippines pitcairn
poland portugal puertorico qatar reunion romania russia rwanda sainthelena
saintkittsnevis saintlucia saintpierremiquelon saintvincentthegrenadines samoa
sanmarino saotomeprincipe saudiarabia senegal seychelles sierraleone slovakia
slovenia solomonislands somalia southafrica southgeorgia srilanka sudan
suriname svalbardjanmayen swaziland syria taiwan tajikistan tanzania thailand
togo tokelau tonga trinidadtobago tunisia turkey turkmenistan
turkscaicosislands uganda ukraine unitedarabemirates uruguay uzbekistan vanuatu
venezuela vietnam virginislandsuk virginislandsus wallisfutuna westernsahara
yemen yugoslavia zambia zimbabwe
primaryclocksource:---> [Shelf {0-255}/Slot {0-31}/Port {0-500}/SubPort/Type] |
[Name/Type]
ringsource:-----> internalringsourcelabel externalringsourcelabel
revertiveclocksource:-> true false
voicebandwidthcheck:-> true false
alarm-levels-enabled:-> critical+major+minor+warning
userauthmode:-----> local radius radiusthenlocal radiusthencraft
radiusauthindex:-----> {0 - 2147483647}
secure:-----> enabled disabled
webinterface:-----> enabled disabled
options:----->
cvlanonly+nol3bridgetable+ipg88bits+disdefpktrules+enablexcardlinkagg+fiberlan+cfmon
reservedVlanIdStart:--> {0 - 4090}
reservedVlanIdCount:--> {0 - 2048}
snmpVersion:-----> snmpv2 snmpv3 snmpv3includingZMS
persistentLogging:----> enabled disabled
outletTemperatureHighThreshold:-> {35 - 65}
outletTemperatureLowThreshold:-> {-40 - 0}

```

To view the card profiles existing on the system, enter **list card-profile**:

```

zSH> list card-profile
card-profile 1/a/10130
card-profile 1/b/10130
card-profile 1/1/10208
card-profile 1/3/10202
card-profile 1/5/10202
card-profile 1/10/10216
card-profile 1/11/10200
card-profile 1/13/10202
8 entries found.

```

To view the **bridge-interface-record** profiles of existing bridges enter **list bridge-interface-record**:

```

zSH> list bridge-interface-record

```

```
bridge-interface-record ethernet2-94/bridge
bridge-interface-record 1-1-1-0-eth-94/bridge
bridge-interface-record ethernet2-220/bridge
bridge-interface-record 1-1-1-0-eth-220/bridge
bridge-interface-record ethernet2-998/bridge
bridge-interface-record 1-1-1-0-eth-998/bridge
6 entries found.
```

### **show command**

Use the **show** command to view all the options in a profile. For example, if you need to find which country codes are available on the MXK, use the **show system** command.

```
zSH> show system
syscontact:-----> {260}
sysname:-----> {260}
syslocation:-----> {260}
enableauthtraps:-----> enabled disabled
setserialno:-----> {0 - 2147483647}
zmsexists:-----> true false
zmsconnectionstatus:--> active inactive
zmsipaddress:-----> {0 - 0}
configsyncexists:-----> true false
configsyncoverflow:---> true false
configsyncpriority:---> none low medium high
configsyncaction:-----> noaction createlist createfulllist
configsyncfilename:---> {68}
configsyncstatus:-----> synccomplete syncpending syncerror syncinitializing
configsyncuser:-----> {36}
configsyncpasswd:-----> {36}
numshelves:-----> {0 - 0}
shelvesarray:-----> {36}
numcards:-----> {0 - 0}
ipaddress:-----> {0 - 0}
alternateipaddress:---> {0 - 0}
countryregion:-----> argentina australia belgium china costarica finland
france germany hongkong italy japan korea mexico netherlands newzealand
singapore spain sweden switzerland uk us afghanistan albania algeria
americansamoa andorra angola anguilla antarctica antiguabarbuda armenia aruba
austria azerbaijan bahamas bahrain bangladesh barbados belarus belize benin
bermuda bhutan bolivia bosniaherzegovina botswana bouvetisland brazil
britishindianoceanterritory bruneidarussalam bulgaria burkinafaso burundi
cambodia cameroon canada capeverde caymanislands centralafricanrepublic chad
chile christmasisland cocosislands colombia comoros congo cookislands
cotedivoire croatia cuba cyprus czechrepublic denmark djibouti dominica
dominicanrepublic easttimor ecuador egypt elsalvador equatorialguinea eritrea
estonia ethiopia falklandislands faroeislands fiji frenchguiana frenchpolynesia
frenchsouthernterritories gabon gambia georgia ghana gibraltar greece greenland
grenada guadeloupe guam guatemala guinea guineabissau guyana haiti
heardislandmcdonaldislands holysee honduras hungary iceland india indonesia
iran iraq ireland israel jamaica jordan kazakistan kenya kiribati northkorea
kuwait kyrgyzstan lao latvia lebanon lesotho liberia libyanarabjamahiriya
liechtenstein lithuania luxembourg macau macedonia madagascar malawi malaysia
```

```

maldives mali malta marshallislands martinique mauritania mauritius mayotte
micronesia moldova monaco mongolia montserrat morocco mozambique myanmar
namibia nauru nepal netherlandsantilles newcaledonia nicaragua niger nigeria
niue norfolkisland northernmarianaislands norway oman pakistan palau
palestinianterritory panama papuanewguinea paraguay peru philippines pitcairn
poland portugal puertorico qatar reunion romania russia rwanda sainthelena
saintkittsnevis saintlucia saintpierremiquelon saintvincentthegrenadines samoa
sanmarino saotomeprincipe saudiarabia senegal seychelles sierraleone slovakia
slovenia solomonislands somalia southafrica southgeorgia srilanka sudan
suriname svalbardjanmayen swaziland syria taiwan tajikistan tanzania thailand
togo tokelau tonga trinidadtobago tunisia turkey turkmenistan
turkscaicosislands uganda ukraine unitedarabemirates uruguay uzbekistan vanuatu
venezuela vietnam virginislandsuk virginislandsus wallisfutuna westernsahara
yemen yugoslavia zambia zimbabwe
primaryclocksource:---> [Shelf {0-255}/Slot {0-31}/Port {0-500}/SubPort/Type] |
[Name/Type]
ringsource:-----> internalringsourcelabel externalringsourcelabel
revertiveclocksource:-> true false
voicebandwidthcheck:-> true false
alarm-levels-enabled:-> critical+major+minor+warning
userauthmode:-----> local radius radiusthenlocal radiusthencraft
radiusauthindex:-----> {0 - 2147483647}
secure:-----> enabled disabled
webinterface:-----> enabled disabled
options:----->
cvlanonly+nol3bridgetable+ipg88bits+disdefpktrules+enablexcardlinkagg+fiberlan+cfmon
reservedVlanIdStart:--> {0 - 4090}
reservedVlanIdCount:--> {0 - 2048}
snmpVersion:-----> snmpv2 snmpv3 snmpv3includingZMS
persistentLogging:----> enabled disabled
outletTemperatureHighThreshold:-> {35 - 65}
outletTemperatureLowThreshold:-> {-40 - 0}

```

Use additional show commands such as **show bridge-interface-record** to view greater detail about bridges.

```

zSH> show bridge-interface-record
vpi:-----> {0 - 4095}
vci:-----> {0 - 65535}
vlanId:-----> {0 - 4090}
stripAndInsert:-----> false true
customARP:-----> false true
filterBroadcast:-----> false true
learnIp:-----> false true
learnUnicast:-----> false true
maxUnicast:-----> {0 - 2147483647}
learnMulticast:-----> false true
forwardToUnicast:-----> false true
forwardToMulticast:-----> false true
forwardToDefault:-----> false true
bridgeIfCustomDHCP:-----> false true
bridgeIfIngressPacketRuleGroupIndex:-> {0 - 2147483647}
vlanIdCOS:-----> {0 - 7}
outgoingCOSOption:-----> disable all

```

```

outgoingCOSValue:-----> {0 - 7}
s-tagTPID:-----> {33024 - 37376}
s-tagId:-----> {0 - 4090}
s-tagStripAndInsert:-----> false true
s-tagOutgoingCOSOption:-----> s-tagdisable s-tagall
s-tagIdCOS:-----> {0 - 7}
s-tagOutgoingCOSValue:-----> {0 - 7}
mcastControlList:-----> {264}
maxVideoStreams:-----> {0 - 1024}
isPPPoA:-----> false true
floodUnknown:-----> false true
floodMulticast:-----> false true
bridgeIfEgressPacketRuleGroupIndex:--> {0 - 2147483647}
bridgeIfTableBasedFilter:-----> none+mac+ip
bridgeIfDhcpLearn:-----> none+mac+ip
mvrVlan:-----> {0 - 4090}
vlan-xlate-from:-----> {0 - 4095}
slan-xlate-from:-----> {0 - 4095}
bridge-type:-----> uplink downlink intralink t1s rlink
pppoa wire mvr user downlinkvideo downlinkdata downlinkpppoe downlinkp2p
downlinkvoice downlinkupstreammcast ipobt1s ipobuplink ipobdownlink

```

### **get command**

Use the **get** command to view the current configuration of profiles. The **get system 0** command displays information on the current MXK system configuration.

```

zSH> get system 0
system 0
syscontact: -----> {}
sysname: -----> {}
syslocation: -----> {}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {false}
zmsconnectionstatus: --> {inactive}
zmsipaddress: -----> {0.0.0.0}
configsyncexists: -----> {false}
configsyncoverflow: ---> {false}
configsyncpriority: ---> {high}
configsyncaction: -----> {noaction}
configsyncfilename: ---> {172.16.160.49_4_1392921484267}
configsyncstatus: -----> {synccomplete}
configsyncuser: -----> {zmsftp}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
numcards: -----> {3}
ipaddress: -----> {172.16.160.49}
alternateipaddress: ---> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: ---> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}

```

```

revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled: -> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {disdefpktrules}
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: ----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}

```

You can find the syscontact information, or whether the MXK is configured to communicate with the Zhone Management System (ZMS — `zmsexists`, `zmsconnectionstatus`, `zmsipaddress`).

### ***update command***

To update the **system 0** profile and all other profiles, use the **update** command. The **update system 0** command walks you through the profile to change specific fields.



**Caution:** You should be very careful when altering profiles. Where available you should use CLI macro commands.

---

For example:

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}: false
...
...
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

### ***delete command***

Use the **delete** command to delete profiles.

```

zSH> delete gpon-traffic-profile 1
gpon-traffic-profile 1
1 entry found.
Delete gpon-traffic-profile 1? [y]es, [n]o, [q]uit : y
gpon-traffic-profile 1 deleted.

```

## Commands: interface show, bridge show

This section describes these commands:

- [interface show command, page 103](#)
- [bridge show command, page 103](#)

### *interface show command*

The **interface show** command displays the numbered or unnumbered (floating) IP interfaces currently available on the MXK.

```
zSH> interface show
1 interface
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 172.24.64.91/24   00:01:47:17:da:0e   ethernet1
-----
```

**Table 7: Interface show column**

Column	Description
Interface	Shows the interface, the card and the physical port of the IP interface.
Status	Shows whether the interface is up or down.
Rd/Address	The IP address assigned to this gateway.
Media/Dest Address	Media/Dest Address is either the MAC address of a device.
IfName	The interface name.

### *bridge show command*

The **bridge show** command displays the bridge interfaces on the MXK. Note that a bridge is a combination of bridge interfaces working together.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls              3105      1/a/5/0/eth      ethernet5/bridge  UP  D 08:00:20:da:77:9c
              D 00:e0:39:ca:04:8e
              D 00:e0:39:98:97:2c
              D 00:60:e0:45:a9:ff
              D 00:50:04:bf:63:48
              D 00:30:48:2e:c8:f2
              D 00:30:19:81:b0:38
              D 00:08:9b:46:9b:26
              D 00:03:e3:97:bb:05
              D 00:03:e3:97:bb:00
              D 00:02:4b:74:d9:e2
              D 00:01:47:5c:34:58
              D 00:01:47:56:75:8e
              D 00:01:47:4e:dc:c0
              D 00:01:47:1a:e4:74
              D 00:01:47:14:c3:00
ipobtls      Tagged 3105      1/a/6/0/ipobridge  ipobridge-3105/bridge  UP  S 00:01:47:11:b7:c6
              S 10.51.5.5
```

2 Bridge Interfaces displayed

Use the **bridge show** command with a VLAN ID to view all the bridges on a VLAN.

```
zSH> bridge show vlan 999
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical          Bridge           St
Table Data
-----
upl           Tagged  999    1/a/3/0/eth      ethernet3-999/bridge
UP S VLAN 999 default
1 Bridge Interfaces displayed
```

Use the **bridge show <bridge interface>** command to view bridge interface information.

```
zSH> bridge show 1/7/3/16/gpononu
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical          Bridge           St
Table Data
-----
dwn           Tg  101/502    1/7/3/16/gpononu  1-7-3-516-gponport-101/bridge
UP D 00:00:ff:00:00:10
dwn           Tg  102/503    1/7/3/16/gpononu  1-7-3-516-gponport-102/bridge
UP
dwn           Tagged  500    1/7/3/16/gpononu  1-7-3-516-gponport-500/bridge
UP
tls           Tagged  848    1/7/3/16/gpononu  1-7-3-516-gponport-848/bridge
UP
dwn           Tagged  998    1/7/3/16/gpononu  1-7-3-916-gponport-998/bridge
UP D 00:21:a1:aa:cd:10
tls           Tagged  2001   1/7/3/16/gpononu  1-7-3-516-gponport-2001/bridge
UP
6 Bridge Interfaces displayed
```

### Commands: bridge stats

You can use the **bridge stats** command to view the packets being sent or received on bridge interfaces. If you add the name of a bridge you can see the statistics for that bridge.

```
zSH> bridge stats
Interface
Counters
Name
Transmitted
ipobridge-3105/bridge
ethernet5/bridge
2 Bridge Interfaces displayed
```

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Received
	UCast	MCast	BCast	Error	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	
ipobridge-3105/bridge	0	0	18	1	16	262	0	0	0	0	0	0	--
ethernet5/bridge	--	--	--	--	--	--	--	--	0	0	0	0	--

## Commands: port show, port up, port down, port bounce, port status

You can use the **port** command to view the administrative state of an interface, change the administrative state of an interface, or change configuration parameters for an interface. See [MXK port management on page 111](#) for more information.

Enter **port show** *<interface>* to view the administrative state of an interface:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location:      1/6/2/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
```

Use port up, down, or bounce to alter the administrative status of a physical or virtual interface. Bounce performs a down operation followed by an up operation.

Enter **port up** *<interface>* to change the administrative state of an interface from down to up:

```
zSH> port up 1-6-2-0/eth
1-6-2-0/eth set to admin state UP
```

Enter **port down** *<interface>* to change the administrative state of an interface from up to down:

```
zSH> port down 1-6-2-0/eth
1-6-2-0/eth set to admin state DOWN
```

Enter **port bounce** *<interface>* to change the interface from UP to DOWN, and back to UP.

```
zSH> port bounce 1-6-2-0/eth
1-6-2-0/eth set to admin state DOWN
1-6-2-0/eth set to admin state UP
```

Enter the **port status** *<interface>* to get the operational status, speed and duplex mode of the Ethernet port.

```
zSH> port status 1-a-1-0/eth
Operational status : Up
Rate in Mbps       : 100
Duplex             : Full
```

## Save and restore configurations

The **dump** command saves the system configuration to the console, a local file, or the network.

The command uses the following syntax:

```
dump [file filename] [network host filename]
```

Passwords are encrypted when they are saved to the configuration file. The encrypted passwords are used to restore the correct password, but cannot be used to log in.




---

**Note:** The **dump** command uses TFTP to transfer files to the network. Set the TFTP server time-out value to at least 5 seconds, and 5 retries to help prevent TFTP timeout or retry errors.

---

## Backing up the configuration to a local file

To dump the configuration to a local file:

Specify a file name for the configuration:

```
zSH> dump file filename
```

The file is saved on the MXK file system.

## Backing up the configuration to the network

To back up the configuration to the network:

- 1 Create the file in the destination location of the TFTP server and make it writeable.
- 2 Save the configuration. The following example saves the configuration to a file named `device.cfg` on the host `192.168.8.21`:

```
zSH> dump network 192.168.8.21 device.cfg
```

## Restoring the configuration

The configuration is restored to the system during **systemreboot**. See [Downloading software files on page 94](#).

## SNTP

### Set system for SNTP

To set up the system to use SNTP update the **ntp-client-config** profile:

```
zSH> update ntp-client-config 0
ntp-client-config 0
Please provide the following: [q]uit.
primary-ntp-server-ip-address: ---> {0.0.0.0}: 192.168.8.100
secondary-ntp-server-ip-address: -> {0.0.0.0}:
local-timezone: -----> {gmt}: pacific
daylight-savings-time: -----> {false}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
```

Record updated.

## Set Daylight Savings Time begin and end times

To set the specific date and time for the beginning and end of daylight savings time add the month, day and time in the **daylight-savings-time-start** and **daylight-savings-time-end** parameters of the **ntp-client-config** profile. Follow the MM:DD:HH:MM (month:day:hour:minute) format.

For example to set the daylight savings time to begin on March 10 at 2am and end on November 3 at 2am, the actual times for 2013 DST, you would update the **ntp-client-config** as shown below.

```
zSH> update ntp-client-config 0

ntp-client-config 0
Please provide the following: [q]uit.
primary-ntp-server-ip-address: ---> {172.16.1.53}:
secondary-ntp-server-ip-address: -> {0.0.0.0}:
local-timezone: -----> {pacific}:
daylight-savings-time: -----> {true}:
daylight-savings-time-start: -----> {03:10:02:00}:
daylight-savings-time-end: -----> {11:03:02:00}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```



**Note:** The **primary-ntp-server-ip-address** parameter must be non-zero to save changes to the **ntp-client-config** profile.



**Note:** When testing this feature, please ensure that there is at least two hours time between the start and end times of the cycle for the feature to operate correctly.

---

## MXK Simple Network Management Protocol (SNMP)

This section describes the following:

- [Create SNMP community names and access profiles, page 108](#)
- [Configure traps, page 109](#)

### Create SNMP community names and access profiles



**Note:** By default, the MXK has a single SNMP community defined with the name **ZhonePrivate**. This community has admin access to the system. Zhone recommends that you configure community names and access profiles to prevent unauthorized access to the system.

The **community-profile** specifies the community name and an access level for SNMP manager to access the system. It can also optionally specify a **community-access-profile** which is used to verify the source IP address of the SNMP manager. The system supports up to 50 different access lists.

The following community access levels are supported:

- **noaccess**—the community has no access.
- **read**—the community has read-only access to the system, with the exception of information in the **community-profile** and **community-access-profile**.
- **readandwrite**—the community has read/write access to the system, with the exception of information in the **community-profile** and **community-access-profile**.
- **admin**—the community has read and write access to the entire system, including information in the **community-profile** and **community-access-profile**. Note that the ZMS requires admin access to manage the system.

#### Create a community profile



**Note:** Configuring a community profile disables the **ZhonePrivate** default community name. If you do change the community name, you must change the name in ZMS or the device will become unmanageable.

The following example defines a community name **public** with read-only privileges:

```
zSH> new community-profile 1
Please provide the following: [q]uit.
community-name: -----> {}: public
```

```
permissions: -----> {read}:
access-table-index: -> {0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

### Create community access profiles

The following example defines a community name **private** with read/write privileges and also creates an access list to verify that the SNMP client attempting to access the MXK is coming from known IP addresses 192.168.9.10 and 192.168.11.12:

First, create an access list for the first IP address:

```
zSH> new community-access-profile 2
Please provide the following: [q]uit.
access-table-index: -> {0}: 1
ip-address: -----> {0.0.0.0}: 192.168.9.10
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Then, create an access list for the second IP address with the same **access-table-index (1)**:

```
zSH> new community-access-profile 3
Please provide the following: [q]uit.
access-table-index: -> {0}: 1
ip-address: -----> {0.0.0.0}: 192.168.11.12
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Finally, create a **community-profile** that specifies the community name, and uses the same **access-table-index (1)** as defined in the two **community-access-profiles** you just created:

```
zSH> new community-profile 4
Please provide the following: [q]uit.
community-name: -----> {}: private ZMS must include this name
permissions: -----> {read}: readandwrite
access-table-index: -> {0}: 1
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

### Configure traps

The **trap-destination** profile defines a trap recipient the MXK will send traps to. To configure a trap destination you need:

- the IP address of the SNMP trap server

- the community name the trap recipient expects

The other parameters in the **trap-destination** profile can be left at their default values. The following example configures a trap recipient with the IP address 192.168.3.21:

```
zSH> new trap-destination 32
Please provide the following: [q]uit.
trapdestination: -> {0.0.0.0}: 192.168.3.21
communityname: ---> {}: public
resendseqno: -----> {0}:
ackedseqno: -----> {0}:
traplevel: -----> {low}:
traptype: -----> {(null)}: 0
trapadminstatus: -> {enabled}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```



**Note:** When ZMS configures a device, a trap destination profile is automatically created.

---

## MXK port management

This section describes port management on the MXK:

- [Port command overview, page 111](#)
- [View the administrative and operational states of ports with the port status and port show command, page 112](#)
- [View DDM data on Ethernet SFPs with the port show command, page 112](#)
- [Change port administrative states with the port testing, up, down, or bounce commands, page 115](#)
- [Port descriptions on the MXK, page 117](#)
- [Port mirroring, page 123](#)
- [Ethernet Jumbo Frames, page 126](#)

### Port command overview

The **port** command has various administrative functions and is used to:

- alter the administrative status of a physical port or virtual interface on the MXK with the **port up**, **port down**, **port bounce**, or **port testing** commands. See [Port descriptions on the MXK on page 117](#).
- verify the administrative status of a physical port or virtual interface on the MXK with the **port show** command. See [View the administrative and operational states of ports with the port status and port show command on page 112](#).
- View DDM data on Ethernet SFPs with the **port show** command. See [View DDM data on Ethernet SFPs with the port show command on page 112](#).
- view the operational status, speed, and duplex mode of Ethernet ports with the **port status** command. See [View the administrative and operational states of ports with the port status and port show command on page 112](#).
- associate a text string with a physical interface, including bond groups, with the **port description** set of commands. See [Port descriptions on the MXK on page 117](#).
- display or clear various statistical information on Ethernet ports with the **port stats** command. See [MX\(P\)-160/260 enhanced Ethernet port statistics on page 369](#).
- set the severity level of alarms on Ethernet ports with the **port config alarm** command. See [Settable alarm severity for Ethernet ports on page 1165](#).

- configure jumbo Ethernet frames with the port config command and verify the change with the port show command. See [Ethernet Jumbo Frames on page 126](#)

## View the administrative and operational states of ports with the port status and port show command

### port status and port show command

Use the **port status** command to view the operational status, speed, and duplex mode of an Ethernet port.



**Note:** The **port status** command is only valid for Ethernet ports.

```
zSH> port status 1-6-1-0/eth
Operational status : Up
Rate in Mbps      : 1000
Duplex            : Full
```

Use the **port show** command to view the administrative status of a port or interface.

```
zSH> port show 1-2-1-0/vdsl
Interface 1-2-1-0/vdsl
  Physical location: 1/2/1/0/vdsl
  Administrative status: up
```

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
  Physical location: 1/a/2/0/eth
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

```
zSH> port show 1-6-1-0-eth/bridge
Interface 1-6-1-0-eth/bridge
  Administrative status: up
```

## View DDM data on Ethernet SFPs with the port show command

This section describes DDM on SFPs for Ethernet:

- [DDM data on Ethernet SFPs overview, page 113](#)
- [DDM data on Ethernet line card Ethernet SFPs, page 113](#)
- [DDM data on uplink card Ethernet SFPs, page 114](#)

## DDM data on Ethernet SFPs overview

Digital Diagnostic Monitoring (DDM) provides SFP diagnostic data. For SFPs that support DDM, the SFP transceiver measures the temperature, supply voltage, transmit bias current, transmit power, and the receive power on the SFP.

Use the **port show interface/type** to display DDM data on Ethernet ports using SFPs that support DDM. [Table 8](#) describes the DDM data fields displayed.

For information on GPON DDM, see [Received Signal Strength Indication \(RSSI\) and Digital Diagnostic Monitoring \(DDM\)](#), page 948.

**Table 8: port show command output fields for DDM data on Ethernet ports**

Field	Description
<b>Temperature</b>	Internally measured Transceiver Temperature in degrees celsius.
<b>Voltage</b>	Internally measured Transceiver Supply Voltage in hundredths of volts.
<b>Tx Bias Current</b>	Measured Tx Bias current in milliamperes.
<b>Transmit Power</b>	Measured Tx Output power in tenths of dB.
Receive Power	Measured Rx power in tenths of dB.

## DDM data on Ethernet line card Ethernet SFPs

SFP supports DDM data on Ethernet line card.

```
zSH> port show 1-1-1-0/eth
Interface 1-1-1-0/eth
  Physical location:    1/1/1/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
  DDM data:
    Temperature:      31c
    Voltage:          3.29v
    Tx bias current:  29mA
    Transmit power:  -2.3dBm
    Receive power:   0.2dBm
```

SFP does not support DDM data on Ethernet line card.

```
zSH> port show 1-2-1-0/eth
Interface 1-2-1-0/eth
  Physical location:    1/2/1/0/eth
  Administrative status: up
```

```
Port type specific information:  
  Frame size: 0 bytes  
  Ingress rate: 0 Kbps burst size: 0 Kbits  
  Egress rate: 0 Kbps burst size: 0 Kbits  
DDM not supported
```

SFP not present on the Ethernet port of the Ethernet line card.

```
zSH> port show 1-1-10-0/eth  
Interface 1-1-10-0/eth  
  Physical location: 1/1/10/0/eth  
  Administrative status: down  
  Port type specific information:  
    Frame size: 0 bytes  
    Ingress rate: 0 Kbps burst size: 0 Kbits  
    Egress rate: 0 Kbps burst size: 0 Kbits  
SFP not present
```

## DDM data on uplink card Ethernet SFPs

Ethernet port on uplink card with SFP that supports DDM data.

```
zSH> port show 1-a-2-0/eth  
Interface 1-a-2-0/eth  
  Physical location: 1/a/2/0/eth  
  Administrative status: up  
  Port type specific information:  
    Frame size: 0 bytes  
    Ingress rate: 0 Kbps burst size: 0 Kbits  
    Egress rate: 0 Kbps burst size: 0 Kbits  
  DDM data:  
    Temperature: 24c  
    Voltage: 3.31v  
    Tx bias current: 27mA  
    Transmit power: -2.1dBm  
    Receive power: 0.1dBm
```

Ethernet port on uplink card with without SFP.

```
zSH> port show 1-a-3-0/eth  
Interface 1-a-3-0/eth  
  Physical location: 1/a/3/0/eth  
  Administrative status: up  
  Port type specific information:  
    Frame size: 0 bytes  
    Ingress rate: 0 Kbps burst size: 0 Kbits  
    Egress rate: 0 Kbps burst size: 0 Kbits  
SFP not present
```

Ethernet port on uplink card with SFP that does not support DDM data.

```
zSH> port show 1-a-5-0/eth  
Interface 1-a-5-0/eth  
  Physical location: 1/a/5/0/eth
```

```

Administrative status: up
Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
DDM not supported

```

Ethernet craft port on uplink card that does not use SFPs.

```

MXK-23> port show 1-a-1-0/eth
Interface 1-a-1-0/eth
    Physical location:    1/a/1/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
No DDM data available from ethernet port

```

## Change port administrative states with the port testing, up, down, or bounce commands

### port testing command

Use the **port testing** command to set the administrative state to testing on an Ethernet port.

```

zSH> port testing 1-6-1-0/eth
1-6-1-0/eth set to admin state TESTING

```

Verify the state.

```

zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
    Physical location:    1/6/1/0/eth
    Description:         Test
    Administrative status: testing
    Port type specific information:
        Link state mirroring not configured.

```

Use the **port testing** command to set the administrative state to testing on an VDSL2 port.

```

zSH> port testing 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state TESTING

```

Verify the state.

```

zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
    Physical location:    1/1/1/0/vdsl
    Administrative status: testing

```

## port up command

Use the **port up** command to set the administrative state to up on an Ethernet port.

```
zSH> port up 1-6-1-0/eth
1-6-1-0/eth set to admin state UP
```

Verify the state.

```
zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
  Physical location:    1/6/1/0/eth
  Description:         Test
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

Use the **port up** command to set the administrative state to up on an VDSL2 port.

```
zSH> port up 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state UP
```

Verify the state.

```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
  Physical location:    1/1/1/0/vdsl
  Administrative status: up
```

## port down command

Use the **port down** command to set the administrative state to up on an Ethernet port.

```
zSH> port down 1-a-2-0/eth
1-a-2-0/eth set to admin state DOWN
```

Verify the state.

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
  Physical location:    1/a/2/0/eth
  Administrative status: down
  Port type specific information:
    Link state mirroring not configured.
```

Use the **port down** command to set the administrative state to up on an VDSL2 port.

```
zSH> port down 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state DOWN
```

Verify the state.

```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
  Physical location:    1/1/1/0/vdsl
  Administrative status: down
```

## port bounce command

Use the **port bounce** command to perform a down operation followed by an up operation on an Ethernet port.

```
zSH> port bounce 1-a-2-0/eth
1-a-2-0/eth set to admin state DOWN
1-a-2-0/eth set to admin state UP
```

Verify the state.

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
  Physical location:    1/a/2/0/eth
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

Use the **port bounce** command to perform a down operation followed by an up operation on a VDSL2 port.

```
zSH> port bounce 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state DOWN
1-1-1-0/vdsl set to admin state UP
```

Verify the state.

```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
  Physical location:    1/1/1/0/vdsl
  Administrative status: up
```

## Port descriptions on the MXK

This section describes port descriptions:

- [Port description rules, page 117](#)
- [Add, modify, list, and delete a port description, page 118](#)
- [Search port descriptions, page 122](#)

## Port description rules

The MXK has a port description field, which provides a mapping between the physical port, or bonded interface, or bridge and a subscriber. This mapping improves MXK management without requiring extra documents to provide the mapping. Port description information can be entered for ports, bridges, or bond groups. Port description information is also searchable.

The rules for entering a port description are:

- Port descriptions do not have to be unique.
- The port description field is a text field 64 characters long.
- Any characters can be used including spaces, \$, @, -, ., etc. The only characters not supported are the double quote, ‘ ‘ which is a delimiter to identify the beginning and end of the text string, the carat ‘^’, and the question mark ‘?’.
- Port descriptions are associated with physical ports and not logical interfaces. For bonding technologies port descriptions are supported both on the physical port and the bond group, so if you want to use a keyword such as a company name to group interfaces.
- Even though port descriptions are searchable, you cannot perform commands using port description. For example, you can not use a command like “bridge modify circuitName ...”

## Add, modify, list, and delete a port description

The **port description add** command associates a text string with a physical interface (which includes bond groups):

```
port description add <physical interface> <text string>
```



**Note:** Port descriptions do not need to be unique. If one customer has many lines, they may all have the same port description. You may also use the port description field as a means to group interfaces. See [Search port descriptions, page 122](#).

### Add a port description to a port

To add a port description with spaces to a port, enter:

```
zSH> port description add 1-6-1-0/eth "510 555 5555"
```

In this case, the port description has spaces so quotes are needed.

To verify the port description, enter:

```
zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
  Physical location:    1/6/1/0/eth
  Description:         510 555 5555
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

To add a port description without spaces to a port, enter:

```
zSH> port description add 1-6-2-0/eth BusinessPark
```

To verify the port description enter:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location: 1/6/2/0/eth
  Description:      BusinessPark
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

### **Add a port description to a GPON OLT port and ONU port**

Both the GPON OLTs and the ONUs can have port descriptions.

To add a port description on a GPON OLT, enter:

```
zSH> port description add 1-4-1-0/gponolt SFO
```

To verify the port description, enter:

```
zSH> port show 1-4-1-0/gponolt
Interface 1-4-1-0/gponolt
  Physical location: 1/4/1/0/gponolt
  Description:      SFO
  Administrative status: up
```

To add a port description to a GPON ONU, enter:

```
zSH> port description add 1-4-1-1/gpononu "business 1 555-555-5555"
```

In this case, a port description is added to ONU 1 on OLT 1.

To verify the port description, enter:

```
zSH> port show 1-4-1-1/gpononu
Interface 1-4-1-1/gpononu
  Description:      business 1 555-555-5555
  Administrative status: up
```

### **Add a port description to a bridge**

The port description must be add to the physical port of a bridge configuration. A port description can be added to the physical port of an existing bridge configuration or the port description can be added to the physical port that is then configured as a bridge.

View existing bridges:

```
zSH> bridge show
  Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data
-----
dwn          200      1/6/2/0/eth  1-6-2-0-eth/bridge
DWN
```

```
upl          Tagged 200 1/a/8/0/eth          ethernet8-200/bridge
DWN S VLAN 200 default
2 Bridge Interfaces displayed
```

Add the port description to the physical port of an existing bridge configuration, in this case the downlink bridge on Ethernet port 2:

```
zSH> port description add 1-6-2-0/eth "US Insurance Consortium, Inc."
```

Verify the port description on the downlink bridge:

```
zSH> bridge showdetail 1-6-2-0-eth/bridge
Bridge interface: 1-6-2-0-eth
  Administrative status: up          Operational status: down
  Blocked status: unblocked
  Type:dwn                          200
  Data:
Physical interface: 1-6-2-0/eth
  Administrative status: up          Operational status: down
Description: US Insurance Consortium, Inc.
Interface On Demand Stats State: enabled
Total Packet Statistics
  Received
    Unicast: 0
    Multicast: 0
    Broadcast: 0
    Bytes: --
  Sent
    Unicast: 0
    Multicast: 0
    Broadcast: 0
    Bytes: --
    Errors: 0
  Packet Storm Blocked
    Unicast: 0
    Multicast: 0
    Broadcast: 0
    Alarms: 0
Delta Packet Statistics - Collecting a 1 second data interval
  Received                               Sent
  Unicast Multicast Broadcast Unicast Multicast Broadcast Error
Delta      0          0          0          0          0          0          0          0
Rate      0          0          0          0          0          0          0          0
  IGMP Received                          IGMP Transmitted
  GenQuery SpecQuery vxReport Leave GenQuery SpecQuery vxReport Leave
  0/0      0/0      0/0      0      0/0      0/0      0/0      0
IGMP misc: unknown= 0 errorRx= 0 actChans= 0 actHosts= 0
```

**Add a port description to a bond group**

View the existing bond groups:

```
zSH> bond show all
Bond Groups
```

Slot	GrpId	Type	State	Name	Desc
5	124	efmbond	OOS	1-5-124-0	-
5	25	efmbond	OOS	bond-0025	-

To add a port description to an existing bond group enter:

```
zSH> port description add bond-0025/efmbond "Mary's Nail Shop"
```

To verify the port description on the bond group enter:

```
zSH> bond show group bond-0025/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
5	25	efmbond	OOS	bond-0025	Mary's Nail Shop

```
Group Members
```

Slot	Port	Type	State	Name	Desc
5	2	shdsl	OOS	1-5-2-0	-
5	4	shdsl	OOS	1-5-4-0	-
5	3	shdsl	OOS	1-5-3-0	-

Or enter:

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
5	124	efmbond	OOS	1-5-124-0	-
5	25	efmbond	OOS	bond-0025	Mary's Nail Shop

### **Modify a port description**

The port description modify command allows you to edit an existing port description.

```
port description modify <physical interface> <text string>
```

Enter a port description:

```
zSH> port description add 1-4-1-2/gpononu "Cafe Roma"
```

Verify the description:

```
zSH> port show 1-4-1-2/gpononu
```

```
Interface 1-4-1-2/gpononu
  Description:      Cafe Roma
  Administrative status: up
```

Modify the description on the same port:

```
zSH> port description modify 1-4-1-2/gpononu "Cafe Barrone"
```

Verify the change:

```
zSH> port show 1-4-1-2/gpononu
```

```
Interface 1-4-1-2/gpononu
  Description:      Cafe Barrone
```

Administrative status: up

### **Port description list**

The **port description list** command will list the descriptions on a particular port.

```
zSH> port description list 1/4/1
Interface                               Description
-----
1-4-1-0/gponolt                         SFO
1-4-1-1/gpononu                         business 1 555-555-5555
1-4-1-2/gpononu                         Cafe Barrone
```

### **Port description delete**

The port description delete command removes the port description from the physical interface.

```
port description delete <physical interface>
```

To view the port description on a physical port enter:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location:    1/6/2/0/eth
  Description:         US Insurance Consortium, Inc.
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
  SFP not present
```

To delete the port description enter:

```
zSH> port description delete 1-6-2-0/eth
```

To verify the deletion enter:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location:    1/6/2/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
  SFP not present
```

### **Search port descriptions**

The **port description find** command provides a textual search which allows you search for a text string within the port description fields. The display