ZyPer

MultiCodec Management Platform

User Manual
Updated September 2025
API Release 4.1x

Safety Instructions

1. Read these instructions.

2. Keep these instructions.

3. Heed all warnings.

4. Follow all instructions.

5. Do not use this product near water.

6. Clean only with a dry cloth.

7. Do not block any ventilation openings. Install in accordance with the manufacturer's instructions.

8. Do not install or place this product near any heat sources such as radiators, heat registers, stoves, or other apparatus (including amplifiers) that produce heat.

9. Do not defeat the safety purpose of a polarized or grounding-type plug. A polarized plug has two blades with one wider than the other. A grounding type plug has two blades and a third grounding prong. The wide blade or the third prong are provided for your safety. If the provided plug does not fit into your outlet, consult an electrician for replacement of the obsolete outlet.

10. Protect the power cord from being walked on or pinched particularly at plugs, convenience receptacles, and the point where they exit from the apparatus.

11. Only use attachments/accessories specified by the manufacturer.

12. To reduce the risk of electric shock and/or damage to this product, never handle or touch this unit or power cord if your hands are wet or damp. Do not expose this product to rain or moisture.

13. Unplug this apparatus during lightning storms or when unused for long periods of time.

14. Refer all servicing to qualified service personnel. Servicing is required when the apparatus has been damaged in any way, such as power-supply cord or plug is damaged, liquid has been spilled or objects have fallen into the apparatus, the apparatus has been exposed to rain or moisture, does not operate normally, or has been dropped.

15. Batteries that may be included with this product and/or accessories should never be exposed to open flame or excessive heat. Always dispose of used batteries according to the instructions.

Operating Notes

- The ZyPer Management Platform includes the ZMP web interface. The following browsers are supported:

  ► Google Chrome version 55.0.2883 or greater

- Refer to the Support page on the ZeeVee web site to download the latest firmware.

Contacting ZeeVee

Support

Contact us for installation and technical support, repairs, and warranty service:

+1 (877) 4-ZEEVEE (1.877.493.3833)

zv_support@kramerav.com

zv_supporteurope@kramerav.com

Sales

North America:

+1 (347) 851-7364 Phone

ZV_sales_team@kramerav.com

EMEA:

+44 1494 956677 Phone

ZV_sales_team@kramerav.com

DACH:

+49 171 3620083 Phone

ZV_sales_team@kramerav.com

Features

- Pre-configured Linux O/S is maintenance-free and includes upgrades and support.
- Plug & Play operation will discover and enable labeling and control of any number of ZyPer4K, ZyPerUHD, ZyPerUHD60 or KDS-17 encoders and decoders.
    - Note: Release 2.3.x was the final release to support the ZyPerHD
- Interface allows the independent routing of video, audio and control signals.
- The feature-rich API makes ZyPer4K / ZyPerUHD / ZyPerUHD60 / KDS-17 the perfect add-on to existing distribution systems without the time and dollars usually required for custom programming.
- Presets enable signal routing and scheduling of saved, pre-defined source-display settings for easy duplication and recall.
- Real time system monitoring includes generating alerts for offline or disconnected ZyPer4K / ZyPerUHD / ZyPerUHD60 / KDS-17 devices, sources and displays.
- Auto detection/discovery of additional encoders and decoders make system scaling a snap.
- Easily create and manage video walls of any pattern or configurations up to a 15 x 15 array.
- Create and manage Multi-view displays with up to 19 sources. (ZyPer4K only)

New in Release 4.1, 4.0 and 3.X

New Features 4.1

- KVM Roaming (ZyPer4K-XS/XSE/XR only)
- AES67 Support (ZyPerUHD60 only)
- Active Directory Lightweight Directory Services (ADLDS)
- Support for Kramer KDS-17 (Compatible with ZyPerUHD60)

New Features 4.0

- Support for ZyPer4K-XSE units added
- Ability to disable Icron USB in ZyPer4K-XSE added to 4.0.3.x
- Zones features updated
- Commands for ZyPer V.A.M. added (Visualization, Analysis, Monitoring Tool) Note this feature is Beta. (Not supported on ZyPerUHD60-2EMP)
- Transport Layer Security updates
- 802.1X Support added (ZyPer4K-XS/XSE/XR only)
- Multicast discovery enabled for ZyPer4K endpoints.
- Update Dante IP address (ZyPerUHD60-2EA and 2DA only)

New Features 3.2

- Added support for ZyPerUHD60 2E, 2EA, 2D, 2DA
- Added specific commands related to Dante enabled UHD60 encoders and decoders

New Features 3.1

- Added support for ZyPer4K 12G SDI Encoder

New in Release 2.3 / 2.4 / 2.5 / 3.0

New Features 3.0

- Many account and security related features
- Flash LED lights from GUI
- Set Low Power mode from GUI
- See Icron USB IP addresses in Display/Source Grid
- Set Server IP address from GUI

New Features 2.3 / 2.4

- Additional Multiview preset patterns. (ZyPer4K only)
- Clock added to Preset Calendar.
- Copy/Clone command added for Multiview. (ZyPer4K only)
- License count only applies to ZyPer4K units. Non ZyPer4K units do not count against license limit.
- Channel up/down command added for Multiview windows. (ZyPer4K only)
- Update ZyPerUHD "No Source Found" background from ZMP GUI.
- Ability to enable or disable viewing of IP address and firmware version in ZyPerUHD "No Source Found" screen. (Release 2.3.37261 and newer)
- Updated help search features for API
- Release 2.4 includes update to Linux version 20.04 on the new NUC-E form factor Management Platform. (See hardware specifications in Section 1)

New Features 2.5

- Update ZyPerUHD "No Source Found" background from ZMP API.
- Maximum supported video wall size increased to 15x15 for ZyPerUHD
- Ability to disable 5V HDMI line on ZyPer4K-XS/XR decoders when no video routed to the decoder.
- Ability to issue a channel up or channel down command to ZyPer4K decoder via a ZeeVee IR remote control or ZyPer Trigger. Requires ZV IR RX unit.

New Features 2.5.3

- Support for the new ZyPerUHD60

ethernetManagementPort changed to utilityPort

The 1Gb port on ZyPer4K units was referred to as the ethernetManagementPort in previous releases of the API. With release 2.3 this is now changed to utilityPort.

New/Updated Commands 4.1 (See API Command Listing)

- add ldap
- convertPid
- create kvm
- delete kvm
- delete ldap
- redundancy add server
- redundancy delete server
- rename kvm
- set device dante
- set kvm
- set ldap mode
- set ldap server
- set ldap type
- set server isolationMode
- show kvm config
- show kvm swapped
- show ldap
- show server managedDevices
- start kvm
- stop kvm

New/Updated Commands 4.0 (See API Command Listing)

- add snmp netNode
- delete snmp netNode
- generate tls csr
- load tls device
- rename zone
- set decoder analogAudioOut
- set decoder danteAudioOut
- set decoder hdmiAudioOut
- set decoder lowLatency
- set device dante ip
- set device optionCard
- set device usbType
- set device utilityPort
- set encoder analogAudioOut
- set server api lineWrap
- set server contact
- set server discoverMode
- set server ftp mode
- set server hostName
- set server location
- set server redundancy
- set server security
- set snmp netNode
- show device names
- show snmp netNode

New/Updated Commands 4.0 (See API Command Listing)

- show server config
- show server info
- show snmp netNode
- show tls device summary
- show tls ca pem
- show tls device pem
- show tls server pem

New/Updated Commands 3.2 (See API Command Listing)

- help
- join
- join dante
- set encoder danteAudioOut
- set decoder danteAudioOut
- set device security (ZyPer4K-XS/XR only)
- set device utilityPort
- set device videoPort

New/Updated Commands 3.0 (See API Command Listing)

- authenticate username
- create account
- delete account
- delete role
- generate tls
- help
- load account
- load tls
- logout
- set account
- set decoder autoAudioConnections hdmiAudioFollowVideo
- set multiview allowMainStream
- set role
- set server security
- set server timezone
- set tls
- show account
- show device config
- show files
- show logs
- show server ip duplicates
- show tls
- sign tls
- troubleReport

Support for older ZMP NUC Devices

- ZMP NUC-D Version devices are not supported for the 4.1 release. These are Intel NUC devices with a single Ethernet port. These devices are running an incompatible version of the Linux Operating System and were last shipped from ZeeVee in early 2022. They look like the image below:

- First generation ZMP NUC devices are not supported with the 3.0 release of ZMP API. These devices are running an incompatible version of the Linux Operating System and were last shipped by ZeeVee back in 2017. These units can be easily identified as they have the brand name "GigaByte" written on the underside of the unit.
- Customers using an older NUC that wish to upgrade to the 4.1 or newer ZMP API release should contact the Kramer sales team (ZV_sales_team@kramerav.com) to purchase an updated ZMP Hardware device.

These API Commands have been removed

- join dante
- set server ssh password
- set device control authorization

Contents

1 Getting Started

Hardware Specifications 2
- Intel NUC 2
- Updated NUC (2022) 3
- Rack Server 4

Installation 5
- Using Windows® 5
- Using OS X® 5
- Sample Application Diagram 6

Management Platform Login 8
Network Configuration 9
- Fixed Mapping using DHCP 9
- Static IP Configuration 10

2 Basic Operation

Accessing ZyPer Management Platform 12
Home Page 13
- Device Status Indicators 14
- Displaying Devices By Status 16
- Displaying Devices By Name 17
- Joining Encoders to Decoders 18
- Video Preview Stream 21

Source/Display Config Page 22
- Configuring Encoders and Decoders 22
- Deleting, Rebooting or Resetting an Encoder or Decoder 29
- Sending a CEC Command 31
- Sending a RS232 String Command 33
- Updating Firmware 35

Configuring KVM Roaming 37
Creating Video Walls 40
- Bezel Compensation 43

Creating a Multiview Screen 45
Creating Zones 49
Presets and Scheduling 52
Source Grid Page 57
Display Grid Page 62
Server Functions Page 67
- Server License 70
- Server Network Config 71
- Server Security 72

TLS 74
Redundancy 78
Accounts 79
- Two-Factor Authentication 81

Roles 87
Help Page 89

3 Advanced Operation

Accessing the API 91
Using Telnet 91
Getting Help 91
Setting the Time Zone 94
EDID Management 95
Auto EDID Mode 95
Using Custom EDID Data 95
Using AJAX/JSON 97
Login Authentication 97
Command Request / Response 98
Fast-Switched vs. Genlocked Mode 100
API Command Listing 101

4 Event Mechanism

ZMP Event Mechanism 367
Telnet Event Session 367
Browser WebSocket 368
API Polling 369
Event List 369

5 SNMP and LLDP Support

SNMP Agent SNMPv3 Account Configuration 375
SNMP Agent SNMPv2 Account Configuration 376
SNMP Client 376
SNMP Notifications 377
More Technical Information on Linux Net-SNMP Package 378

6 TLS

TLS Support 380
TLS Tests 380

7 802.1X

ZyPer4K-XS/XSE and ZyPer4K-XR 802.1X Support 388

8 Active Directory Lightweight Directory Services

Configuring LDLDS Attributes for ZMP 396
Configuring LDLDS Attributes for SSSD Integration on Linux 398

9 V.A.M. BETA

Visualize, Analyze, Monitor 400

10 Appendix

Updating the Software 412
Using Mac OS X 414
Using Windows 415
Using ZyPer Management Platform 415
Redundancy Configuration Instructions 417
Hardware Specifications 420

1 Getting Started

Hardware Specifications (Older Intel NUC-D version)

CPU	• Intel® Pentium® Processor J5005
Operating System	• Linux Ubuntu 16.04
Internal Storage	• 64 GB SSD
Graphics	• Intel® HD Graphics 600
LAN	• Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 19V DC, 3.42 A
I/O	• 2 x HDMI 2.0a • 4 x USB 3.0, Type- A, female • 1 x RJ45 • 1 x 19V DC • 1 x Kensington lock slot • 2 x 3.5mm headset jacks (Not used)
Operating Temperature	• 0 ºC to +40 ºC
Storage Temperature	• -20 ºC to +60 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 4.55 in x 2.01 in x 4.57 in (115 mm x 51 mm x 111 mm)

Hardware Specifications (NUC-E version Feb 2022 and beyond)

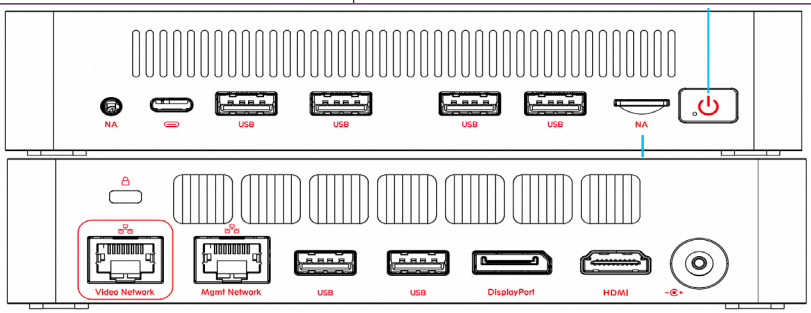
CPU	• Intel® Celeron® Processor N3350
Operating System	• Linux Ubuntu 20.04
Internal Storage	• 64 GB SSD
Graphics	• Intel® HD Graphics 500
LAN	• 1x 100mb LAN (Eth0), 1x Gigabit LAN (Eth1)
Internal Memory	• 4 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 19V DC, 3.42 A
I/O	• 2 x HDMI 2.0a • 3 x USB 3.0, Type-A, female • 2 x USB 2.0, Type-A, female • 2 x RJ45 • 1 x 19V DC • 1 x Kensington lock slot • 1 x 3.5mm headset jacks (Not used)
Operating Temperature	• 0 ºC to +40 ºC
Storage Temperature	• -20 ºC to +70 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 6.06 in x 1.25 in x 4.25 in (154 mm x 32 mm x 108 mm)

RoHS

Ethernet Port 0 = Video Port. DHCP default IP Address (Side with USB only)
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port. (Side with HDMI ports)
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

Hardware Specifications (NUC-F version Q4 2025 and beyond)

CPU	• Intel® Alder-N Lake N100
Operating System	• Linux Ubuntu 22.04
Internal Storage	• 128 GB SSD
Graphics	• Intel® UHD Graphics
LAN	• Dual Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 12V DC, 3A
Power Use	• 24W Peak, 18W Normal operation
I/O	• 1 x HDMI 2.0a • 1 x DisplayPort • 4 x USB 3.0, Type-A, female • 2 x USB 2.0, Type-A, female • 2 x RJ45 • 1 x 12V DC • 1 x Kensington lock slot • 1 x 3.5mm headset jack (Not used) • 1 x MicroSD Card Slot (Not used)
Operating Temperature	• 0 ºC to +45 ºC
Storage Temperature	• -20 ºC to +70 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 6.85 in x 1.42 in x 5.04 in (174mm x 36 mm x 128 mm)



Ethernet Port 0 = Video Port. DHCP default IP Address
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port.
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

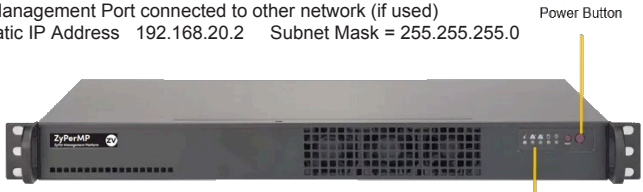
Important Note: The Power/Reset button will reset the unit when button is released. Not when it is pressed. Do not press and hold the button in. Just give a quick press.

Hardware Specifications (Enterprise Grade Rack Mount)

CPU	• Intel® Xeon E3-1200 v5
Operating System	• Linux Ubuntu 16.04 or 22.04 (Started shipping July 2025)
Internal Storage	• 64 GB SSD
Graphics	• ASPEED AST2400 BMC
LAN	• Dual Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• 200W Low-Noise AC-DC power supply. Fan speed dynamically adjusts for load and environment.
I/O	• 1 x VGA (15-pin D-sub) • 2 x USB 2.0, Type-A, female • 2 x RJ45 (LAN) (Video Network and Management Network) • 1 x RS232 (9-pin D-sub)
Operating Temperature	• +10 °C to +35 °C
Storage Temperature	• -40 °C to +70 °C
Dimensions (W x H x D)	• 17.2 in x 1.7 in x 11.3 in • (437 mm x 43 mm x 287 mm)
Ambient Noise	• Measurement point was 1M distant, straight in front of unit • 43.5 db(A) - Startup and peak load condition • 32.8 - 34.4 db(A) - Expected range during typical load • 31.7 db(A) - Idle and very lightly loaded or cooler ambient conditions
Weight	• 8.45 lbs, (3.83 kg)

RoHS

Ethernet Port 0 = Video Port. DHCP default IP Address
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port.
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0



Installation

1. Connect the included power supply to the power receptacle on the ZyPer Management Platform.

2. Connect the included AC power cord from the power supply to an available electrical outlet.

3. Connect an Ethernet cable from the ZyPer Management Platform to a switch that is on the same LAN that will be hosting the ZyPer devices. Although the ZyPer Management Platform can be connected anywhere on the LAN, it is recommended that it is connected to the primary switch where the ZyPer endpoints are connected.

   NOTE: If the ZyPer Management Platform does not detect a DHCP server within 60 seconds, a link-local address of `169.254.xxx.xxx` will be assigned to the ZyPer Management Platform. If you wish to use static or fixed-mapping using DHCP, then see Network Configuration (page 10).

Using Windows®

a. Connect an Ethernet cable from your computer to the same switch as above.

b. Go to the Network folder.

c. Locate the icon titled "ZyPer Management Server" followed by an IP address. This is the IP address of the ZyPer Management Platform.

Using OS X®

a. Connect an Ethernet cable from your computer to the same switch as above.

b. Open the Terminal application.

At the prompt, enter `ping zyper.local.`

c. The IP address of the ZyPer Management Platform will be displayed.

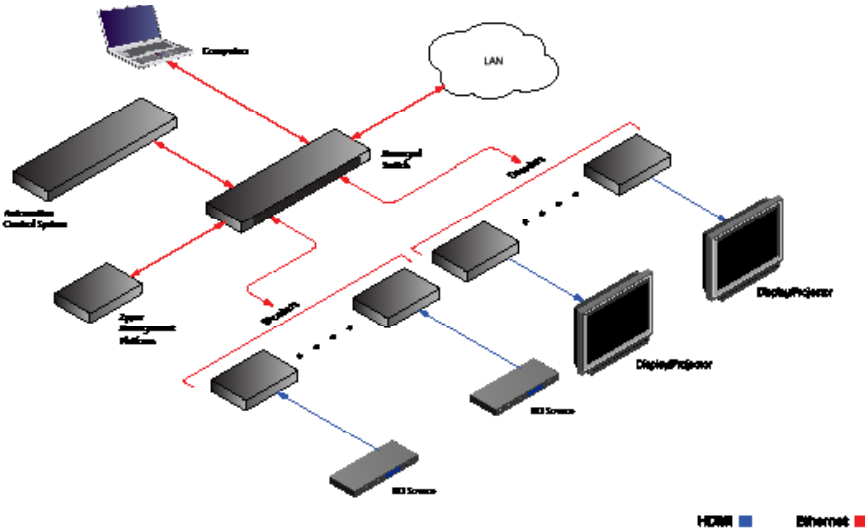
4. Open a web browser and enter the IP address of the ZyPer Management Platform.

5. The login screen for the ZMP will be displayed.

6. Enter `admin` for both the Username and Password fields.
   See Management Platform Login (page 9) for more information.

7. Click the Login button.

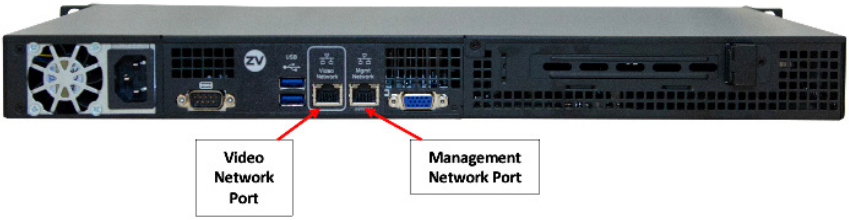
Sample Application Diagram



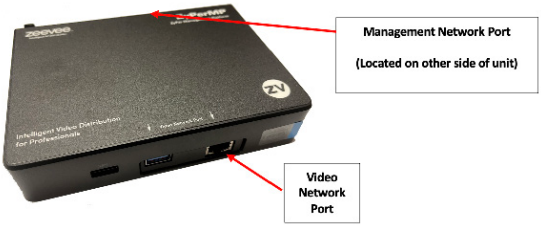
Management Platform Network Ports

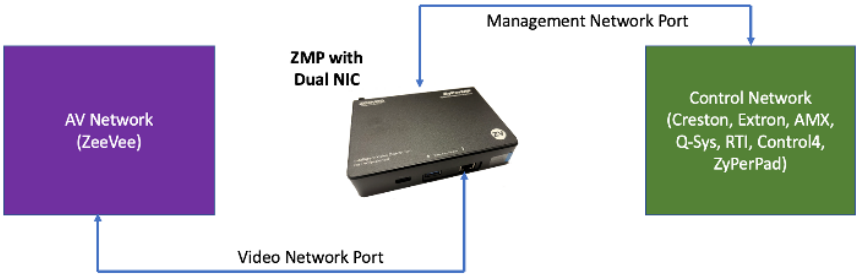
The ZyPer Management Platform (ZMP) is available from ZeeVee in two different hardware options. NUC and Rack Mount Server. Both versions have two network interfaces. It is important that the port labeled "Video Network" is connected to the LAN containing the ZyPer4K/ ZyPerUHD60/ZyPerUHD/KDS-17 encoder/decoder endpoints. The port labeled "Management Network" is available to separate the Video AV Network from a corporate or Control Network.

Rack Mount ProServer ZMP



NUC ZMP





Management Platform Login

1. Directly connect an HDMI cable (NUC) or VGA cable (Rack Mount) between the Management Platform and a Display. (Note that a USB keyboard is also required to be connect to the Management Platform)

2. After boot the login prompt will appear. At the login prompt, enter the following login name: `zyper`

   After entering the login name, the password prompt will be displayed.

   ```
   zyper login: zyper
   Password:
   ```

3. Enter the password. The default password is `zyper`. Note that the password will not be echoed to the screen. Once the password is entered, the screen will appear similar to the following:

   ```
   zyper login: zyper
   Password:
   ```

4. Use the show server info command to find the IP Address of the Managment Platform

```
show server info
server(172.16.6.111);
  server.gen; hostname=zyper.local, serverType=ProServer,
version=4.1.0.41297, previousVersion=4.1.0.41277, master=true,
 location=Art's demo room, contact=Art Weeks
  server.gen; uptime=21d:0h:9m:21s, freeMem= 6.81GB,
 sdvoeVersion=3.6.0.1, bootCount=76, serialNumber=ZZM8M900001A
  server.gen; macAddress=3c:ec:ef:3f:4e:b6, managementMacAddress=3c
:ec:ef:3f:4e:b7
  server.ipActive; ipServerAddr=172.16.6.111,
ipManagementAddr=0.0.0.0, gatewayAddr=172.16.6.1,
dnsAddr=172.16.2.2
  server.ipActive; managementGatewayAddr=0.0.0.0, m
anagementDnsAddr=0.0.0.0
  server.time; time="Wed Jun 25 13:17:04 2025", timezone=US/Eastern
  server.license; productID=4B234600-B306-11EA-8000-3CECEF3F4EB6,
license=none
  server.license; Zyper4KLimit=48, Zyper4KDevices=16,
allDevices=26, allDevicesUp=25, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_2.1.0.9=8, num_2.3.1.1=1, n
um_2.3.1.2=1, num_4.1.2.9=8, num_5.8=3, num_unknown=4
lastChangeIdMax(304);
Success
```

Network Configuration

The default configuration of the Management Platform will use DHCP with link-local addressing support. Link-local addressing allows the Management Platform to have an IP address on a network, even if the Management Platform has not been manually configured or automatically configured by a DHCP server. If a DHCP server is not detected within 60 seconds, a link-local address of `169.254.xxx.xxx` will be assigned to the Management Platform.

If you wish to use a static IP address, this can be done in one of two ways: Add a fixed mapping to the DHCP server or by directly assigning a static address to the Management Platform. Both methods are covered in this section.

Fixed Mapping using DHCP

Before continuing, make sure that the static IP address being used does not conflict with any DHCP-assigned addresses. Contact your system administrator for assistance.

1. At the login prompt, enter the following login name: `zyper`

   After entering the login name, the password prompt will be displayed.

```
zyper login: zyper
Password:
```

Enter the password. The default password is `zyper`. Note that the password will not be echoed to the screen. Once the password is entered, the screen will appear similar to the following:

```
Zyper$
```

Use the show server info command to find the Mac Address of the Management Platform

```
show server info
server(172.16.6.111);
  server.gen; hostname=zyper.local, serverType=ProServer,
version=4.1.0.41297, previousVersion=4.1.0.41277, master=true,
 location=Art's demo room, contact=Art Weeks
  server.gen; uptime=21d:0h:9m:21s, freeMem= 6.81GB,
 sdvoeVersion=3.6.0.1, bootCount=76, serialNumber=ZZM8M900001A
  server.gen; macAddress=3c:ec:ef:3f:4e:b6, managementMacAddress=3c
:ec:ef:3f:4e:b7
  server.ipActive; ipServerAddr=172.16.6.111,
ipManagementAddr=0.0.0.0, gatewayAddr=172.16.6.1,
dnsAddr=172.16.2.2
  server.ipActive; managementGatewayAddr=0.0.0.0, m
anagementDnsAddr=0.0.0.0
  server.time; time="Wed Jun 25 13:17:04 2025", timezone=US/Eastern
  server.license; productID=4B234600-B306-11EA-8000-3CECEF3F4EB6,
license=none
  server.license; Zyper4KLimit=48, Zyper4KDevices=16,
allDevices=26, allDevicesUp=25, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_2.1.0.9=8, num_2.3.1.1=1, n
um_2.3.1.2=1, num_4.1.2.9=8, num_5.8=3, num_unknown=4
lastChangeIdMax(304);
Success
```

After programming the DHCP server to assign a specific address to the Management Platform, reboot the Management Platform, using the following command, to use the new IP address.

```
shutdown server reboot

Success
```

Static IP Configuration

Before continuing, make sure that the static IP address being used does not conflict with any DHCP-assigned addresses. Contact your system administrator for assistance.

The `set server ip` command can be used to set the IP Address of the Management Platform. Refer to API Command Listing (page 94) for a full listing of available commands.

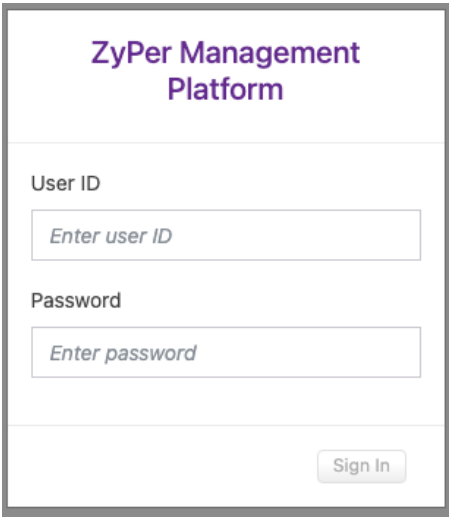
```
set server ip videoPort static address 172.16.16.111 mask
255.255.255.0 gateway 172.16.6.1 dns none reboot

Success
```

2 Basic Operation

Accessing ZyPer Management Platform

1. Open a web browser and enter the IP address of the Management Platform.

2. The login screen for ZMP will be displayed.



3. Enter the required information in the Username and Password fields. The default username and password is `admin`. The username and password are case-sensitive. The “admin” password may be changed by the user at any time.

4. Click the Sign In button.

5. The Home page will be displayed. See the next section for more information.

Home Page

The Home page of the ZyPer Management Platform displays all available Sources, Displays, Walls, Zones, Multiviews, Server Info, Users, Roles, Logs and Help.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. The Home page will be displayed. The Sources pane displays all available encoders . The Displays pane displays available decoders. V.A.M, KVM Config, Walls, Multiview, Zones, Presets, Source Grid, Display Grid, Server, TLS, Redundancy, SNMP Users, Accounts and Roles will be covered in upcoming sections.

Device Status Indicators

Each Source and Display contains a status indicator border color, displaying current information about the device. This is particularly useful for devices that may be in a separate part of a building or several miles away.

Meaning	Indicator Color
OK	Green

1. Indicates that an HDMI cable is connected between the encoder and the source or between the decoder and a display or other sink device. Decoder is receiving a valid video stream.
2. The encoder / decoder is powered.
3. The Ethernet cable is connected between the switch and the encoder / decoder.

Meaning	Indicator Color
Warning	Yellow

1. HDMI cable may not be connected between the encoder and the source or between the decoder and a display or other sink device. Decoder may not be receiving a valid video stream.

   Note that this indicator may also indicate a faulty HDMI cable.
2. The encoder / decoder is powered.
3. The Ethernet cable is connected between the switch and the encoder / decoder.

Meaning	Indicator Color
Error	Red

1. The Ethernet cable is disconnected between the switch and the encoder / decoder.

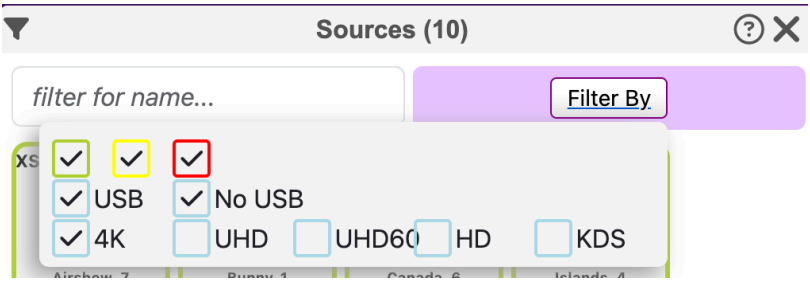
2. The encoder / decoder may not be powered.

Displaying Devices By Status

Both the Sources and Displays window contain a Filter by status option. There are check boxes to filter by Status (Green, Yellow, Red), Device Type (4K, U60, UHD, KDS or HD), USB (Present, Not present) There is also an option to filter Sources or Displays by name.

1. Click the Filter Icon.

2. Only those devices with the selected status will be displayed. As illustrated in this example, only ZyPer4K devices are shown in the Sources window. ZyPerUHD, ZyPerUHD60, ZyPerHD and KDS sources are filtered out.

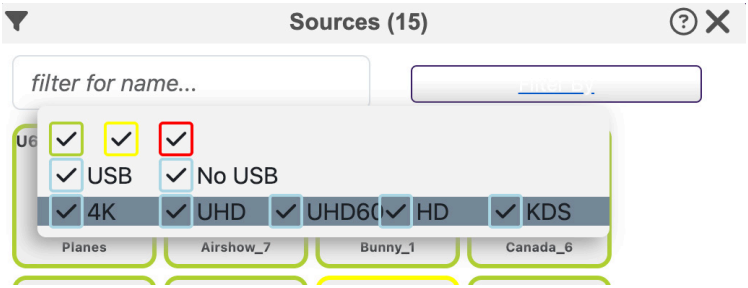


3. Select `every box` from the Filter to show all devices.

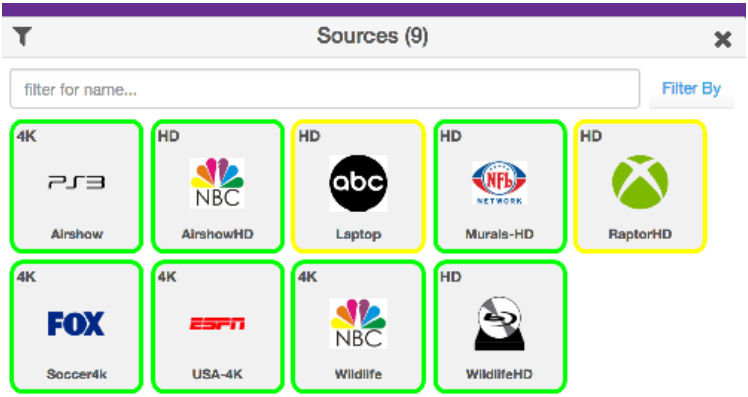
Displaying Devices By Name

Both the Sources and Displays windows contain a Filter by name field. Use this field to enter the name of the desired device(s) to be displayed.
As text is entered, the interface automatically begins a search of the current string for each recognized device. Devices that are displayed must contain the text (in sequence) that is currently in the Filter by name field. Text searches are case-sensitive.

1. Click in the Filter by name field.



This example uses the following named sources. For more information on naming sources see Configuring Encoders and Decoders (page 23).

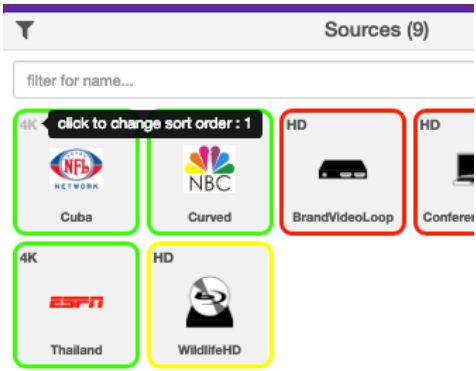


2. Type the desired sequence of characters in the Filter by name field to search. In this example, we want to only display the "Wildlife" source devices. To do this, we can enter part of the name, such as "Wild" or even "W" (since no other device name contains the character "W"). All text entries are case-sensitive.

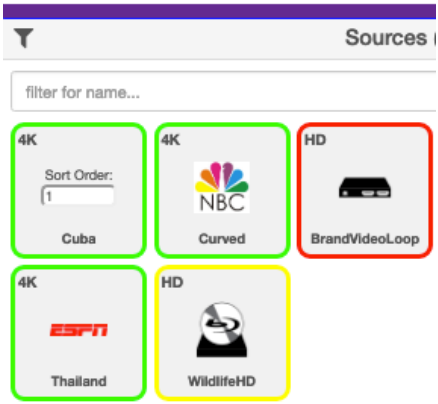
Manually Setting Sort Order

Both the Sources and Displays windows can be sorted manually. Every device is assigned a "sort order" number. The user can change this sort order number manually.

1. Click on the device identifier. (Little "4K" or "UHD" or "HD" in upper left corner of device.) Note: Must be configured for Join Mode rather than Preview Mode. See the section on Video Preview mode on the following pages.



This will bring up a Sort Order option for the selected device. The example below the device is assigned to Sort Order position 1. (Top Left of the Source window) This number can be changed to any number desired up to the number of sources.



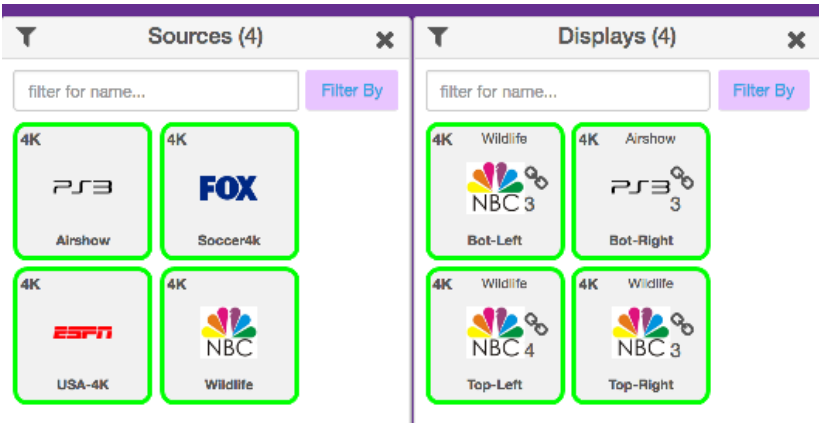
2. Sort Order numbering runs from top left to bottom right. In the image above assuming only 5 sources, the top three would be location 1, 2 and 3; while the bottom two would be locations 4 and 5.

Joining Encoders to Decoders

"Joining" is the process of assigning an encoder (source) to a decoder (display) or a video wall. Before starting the join process, we recommend that you configure the encoder and decoder settings. Refer to Configuring Encoders and Decoders (page 23).

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Drag and drop the desired source on to the desired display.



3. The display icon will change to show that it has now been joined with a source. (Small chain link icon) Hover over the chain link for additional status info. Also the icon will change to match the source and name of the source will appear at the top of the icon.

4. Continue the join process as desired. Note that joining an encoder with a decoder that is already joined, will replace the previous join operation.

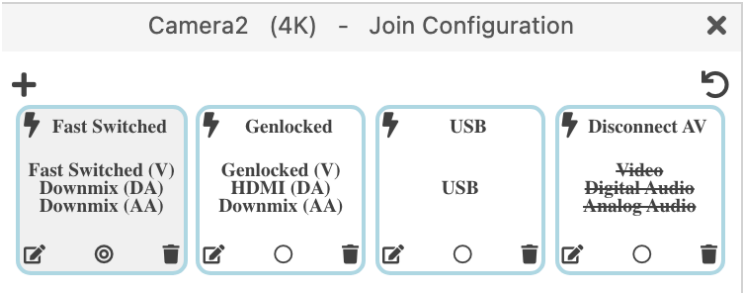
   The join command can also be used to perform the same operation. See API Command Listing (page 94) for more information.

Unlinking Encoders and Decoders

As mentioned earlier, joining an encoder with a decoder that is already joined, will replace the previous join operation. However, there may be situations where you want to completely remove any source from being shown on a display. To do this, use the Disconnect AV feature or Disconnect Joins.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

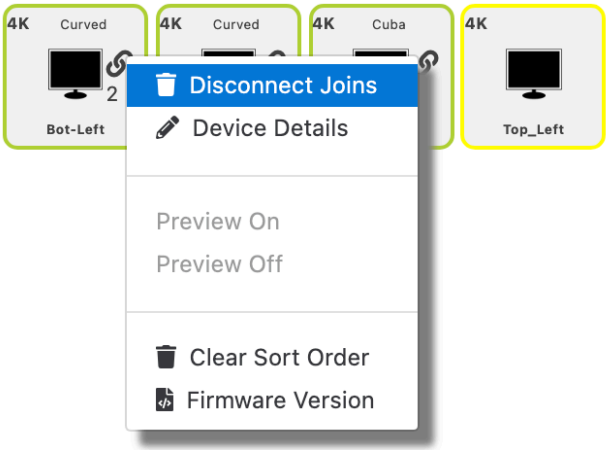
2. On the Sources window, right-click on any Source. Select Configure Join and then Disconnect AV.



3. Drag that source onto the Display that you would like to unlink

4. Alternately you can right click on the "Chain Link" icon and select "Disconnect Joins"

5. The display icon will change to show that it has been unlinked and no longer joined with any encoder. (Yellow perimeter, small chain link icon is gone, icon returns to default and name of joined source is gone)



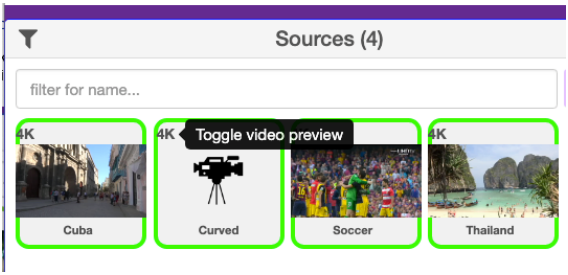
Video Preview Stream

It is possible to view a small thumbnail preview of the active stream in the GUI within either the encoder or decoder box. Once preview streams are enabled, you can see them by following the steps below: (Supported on ZyPer4K, ZyPerUHD60, ZyPerUHD and KDS-17 only) Not supported on ZyPer4K-XS/XSE or ZyPer4K-XR units.

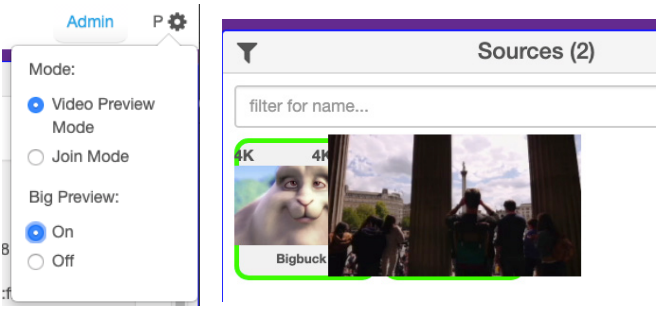
Note: It will take several seconds for the preview stream to appear once activated. There is a maximum of 20 preview streams available at any given time.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. On the Sources window, click on the small "4K" or "UHD" located in the upper left corner of the window. This will toggle the GUI between showing an icon and showing the preview stream. Note: Video Preview Mode but be set at shown in step 3 below.



3. Click on the gear icon in the top right corner of the ZMP to enable or disable the "Big Preview" option. When enabled, hovering the cursor over a preview stream will make the preview approximately 50% larger while you keep the cursor over the stream.



4. Note that preview stream can be viewed for both Sources and joined Displays. All other functionality remains the same. The preview stream updates about 1 time per second. (Multiviews are not available as preview streams. Clicking on preview at the decoder for a multiview will give the appearance that the multiview has failed.)

5. Important Notes: The PC/Laptop attempting to view preview streams MUST have access to the Internet to download a player in the background. ZyPer4K MUST be on firmware version 4.0.1 or newer.

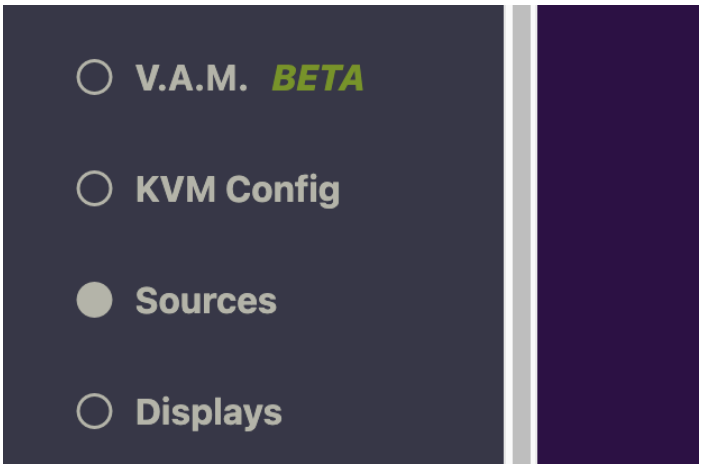
Source/Display Config Page

Configuring Encoders and Decoders

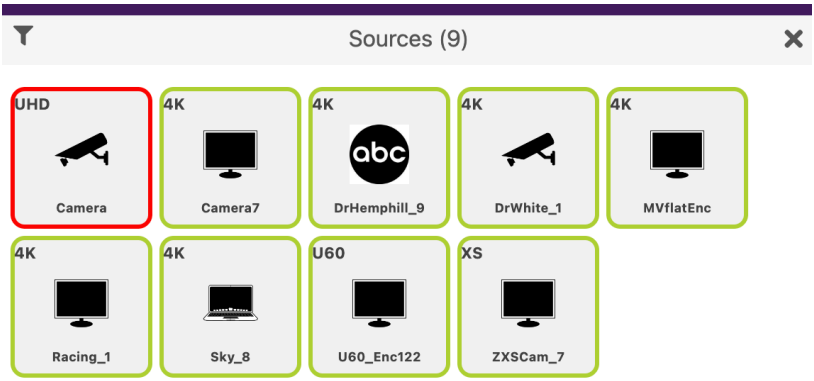
When an encoder or decoder is connected to the network, the Management Platform identifies each unit by its MAC address. This is the default setting. However, when dealing with several units, it is much easier to identify a unit by a string name. It is also possible to assign a preset image to each icon, set the network mode, RS232 settings, and more.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

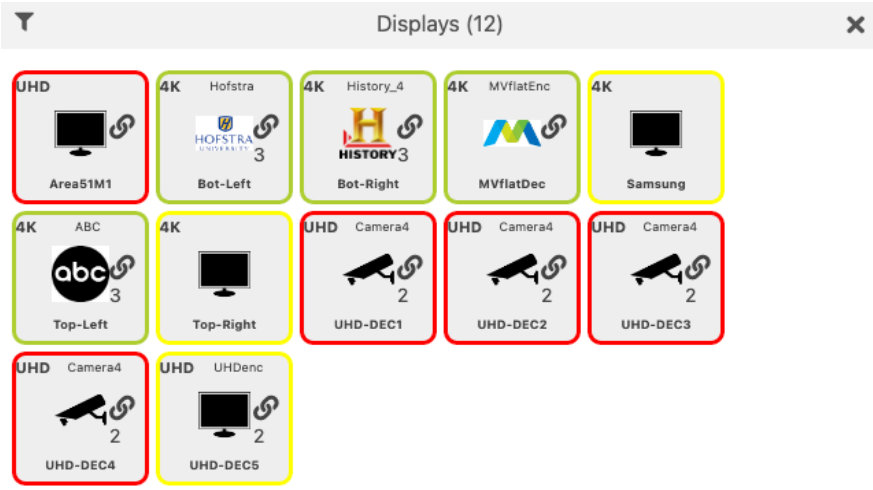
2. Click the Source or Displays tab at the left of the page. Joining Encoders to Decoders (page 20)



3. All available encoders will be displayed under the Sources window. (Maximum 25 per page. Deployments with more than 25 encoders will cycle thru pages)



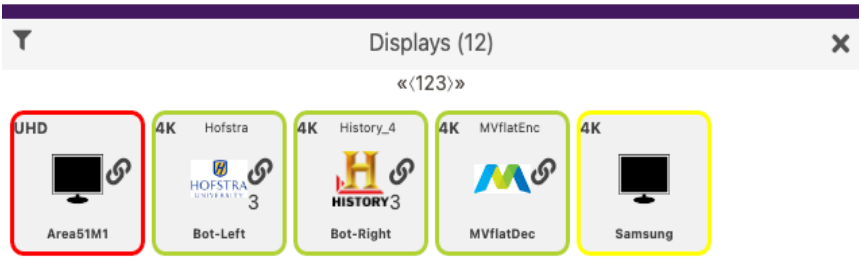
All available decoders will be displayed under the **Displays** window. (Maximum 25 per page. Deployments with more than 25 decoders will cycle thru pages)



Pagination: Note that you can control the number of Sources or Displays shown at one time via the pagination option in the browser:

Add ?max=5 to the end of the brower address to set max Sources/Displays to 5. Valid numbers are 1 thru 25.

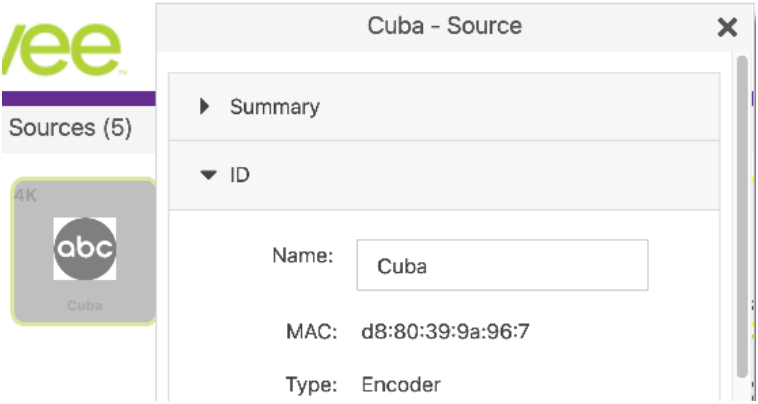
Example: http://172.16.6.111/maestroz/app/?max=5



4. Left-click the desired encoder name to display the context menu. In the example below,we will select the encoder named "Cuba".

5. A menu will appear with options for Summary, ID, Status,Config and Actions.

6. Selecting the ID option will allow you to manually give the Source a name.



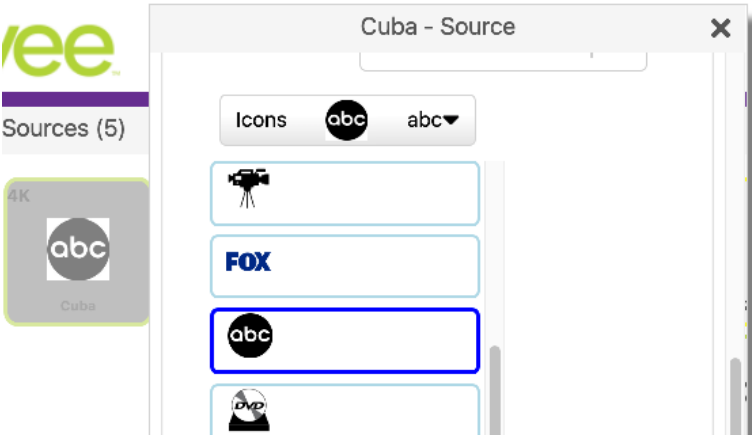
7. Enter a description for the source in the Name field. By default, the Management Platform will use the MAC address of the encoder. This field cannot be blank. Names cannot contain spaces. In this example, we will use "ESPN1".

   Naming a device can also be done using the `set device general name` command. See API Command Listing (page 101) for more information.

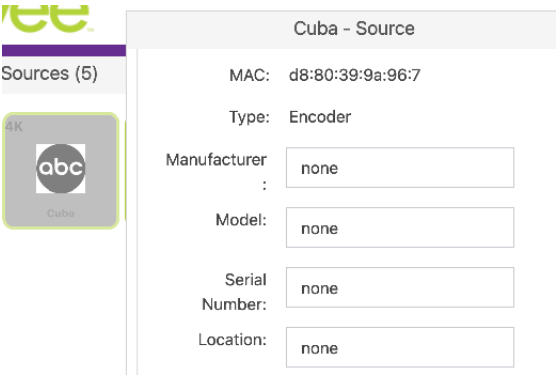
Each encoder and decoder must have a unique name. Duplicate names are prohibited. It should be noted that assigning a name to an encoder does not actually replace the MAC address. The Name field provides a descriptive identifier, rather than a MAC address.

8. Click the Config-Icon drop-down list to select the desired icon for this source. This is optional. If no icon is selected, then the default icon will be used. In this example, we will select **ABC**, since a ABC is connected to this encoder.

   The `set device source-display iconImageName` command can also be used to assign an icon to a device. See API Command Listing (page 101) for more information.



9. Enter the Manufacturer, Model, and Serial Number of the source or display in the appropriate fields. By default, these fields are set to "none". These fields must not be blank. If a custom value is provided, it must not contain spaces.



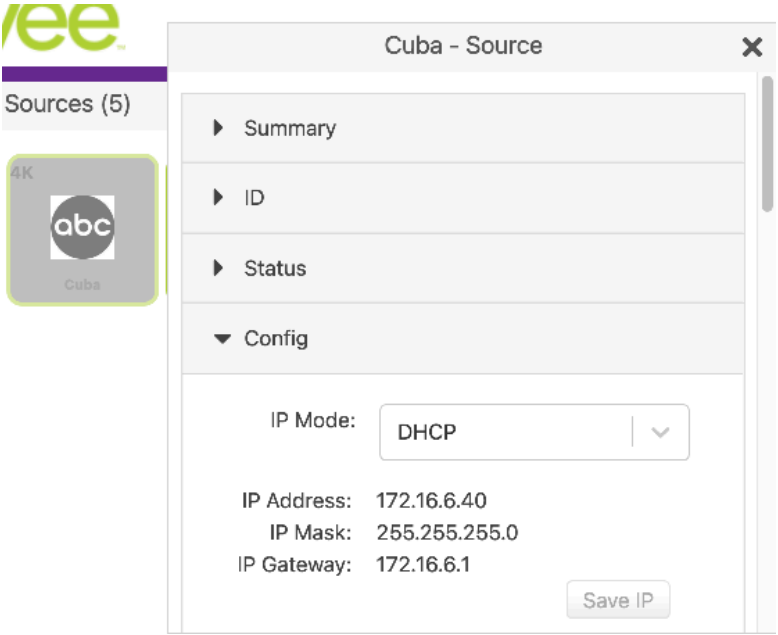
The Manufacturer, Model, and Serial Number can also be assigned using the following commands, respectively. See API Command Listing (page 101) for more information.

- ▶ `set device sourceDisplay manufacturer`
- ▶ `set device sourceDisplay model`
- ▶ `set device sourceDisplay serialNumber`

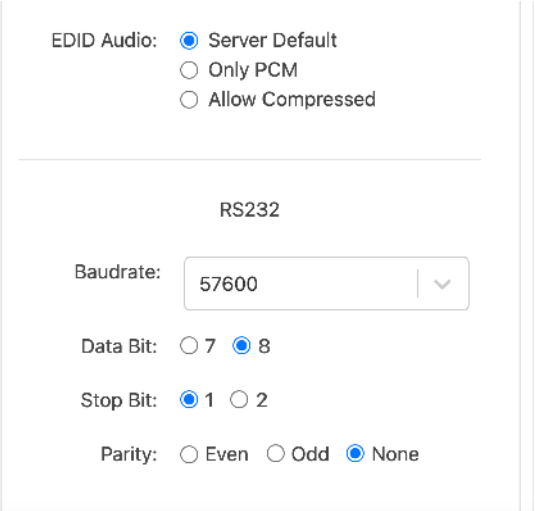
10. Enter the location of the source or display device in the Location field. By default, this field is set to "Unknown". This field must not be blank. If a custom value is provided, it must not contain spaces.

11. The `set device sourceDisplay location` command can also be used to set these values. See API Command Listing (page 101) for more information for more information.

12. Use the Config-IP Mode section to configure the IP setting for the encoder. By default, both encoders and decoders are set to DHCP mode and will be discovered automatically by the Management Platform. To manually configure the IP settings of the encoder, click the Mode drop-down list and select **`static`**. Once in static mode, the information in the Address, Mask, and Gateway fields can be edited.



13. Use the Baudrate section of Config to configure the RS232 settings for the control device, such as an automation control system. Click the Baudrate drop-down list to select the desired baud rate of the control device.

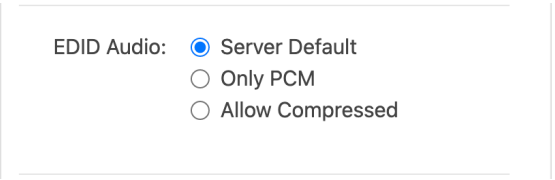


The Device IP settings can also be assigned using the `set device ip dhcp` and `set device ip static` commands. When assigning RS232 settings from the command line, use the `set device rs232` command. See API Command Listing (page 101) for more information.

14. By default, ZyPer4K and ZyPerUHD Encoder audio format parameters will inherit the Server Default configuration.

Only PCM - Ignore the Server Default setting and restrict the EDID to PCM audio only
Allow Compressed - Ignore the Server Default setting and allow Compressed audio on the encoder.

NOTES: For this setting to work properly, the source's audio configuration should be set to automatically determine audio, if possible. If the source is not able to automatically determine audio, it needs to be manually set to the desired auto format. Downmixing hdmi audio (sending audio out of the analog port) can only be done if the hdmi stream is PCM – not compressed.

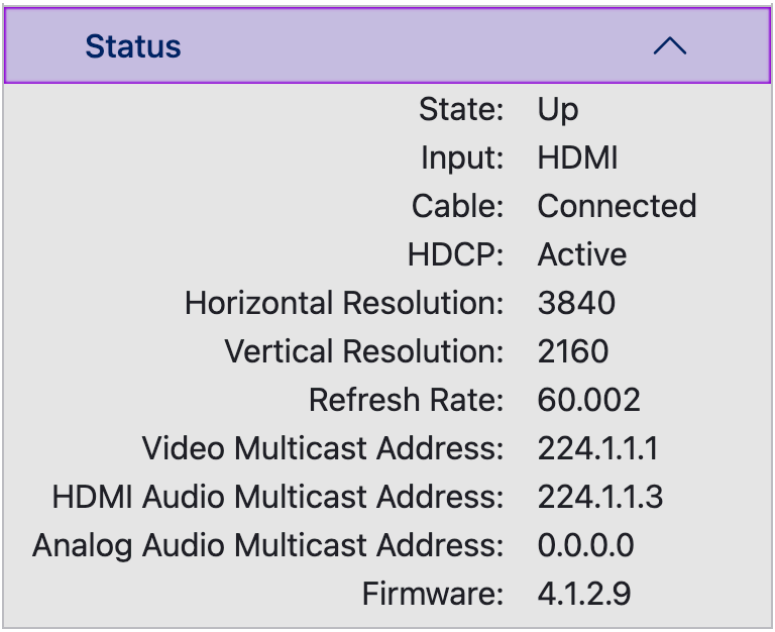


Encoder and Decoder Status Information

You can obtain status information about an encoder and its source at any time, from the Source Config page.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Left-click the desired source name and select the Status option from the context menu.

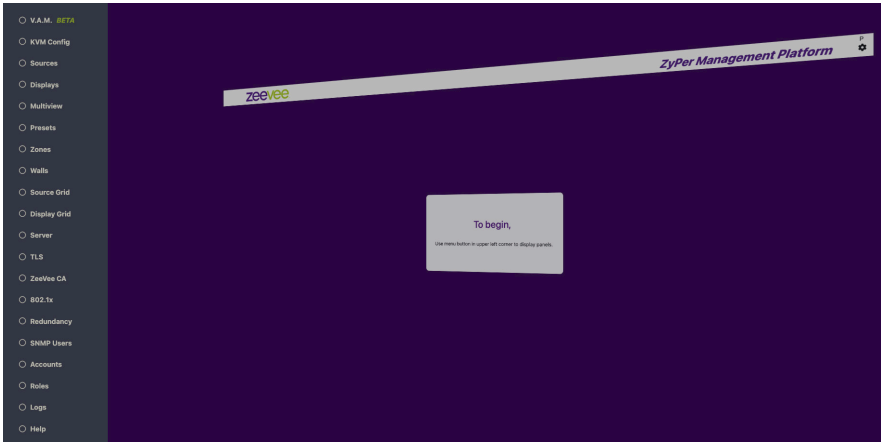


Deleting, Rebooting or Resetting an Encode or Decoder

If an encoder or decoder is disconnected from the network, the Management Platform will continue to display the encoder or decoder within the ZMP until it is removed. Note that; reconnecting the encoder or decoder will cause it to once again be displayed in the ZMP.

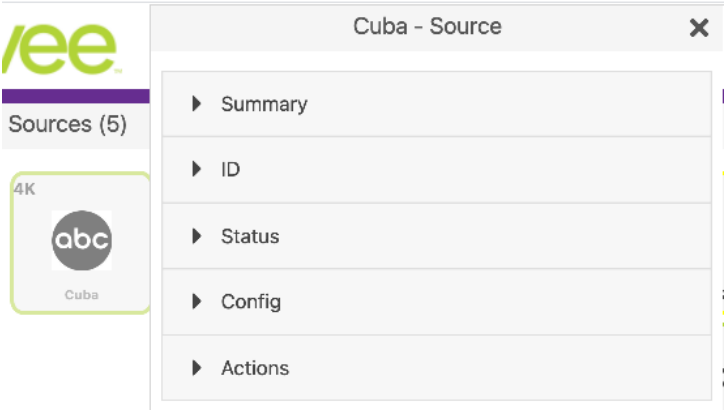
1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Sources tab at the left of the page.

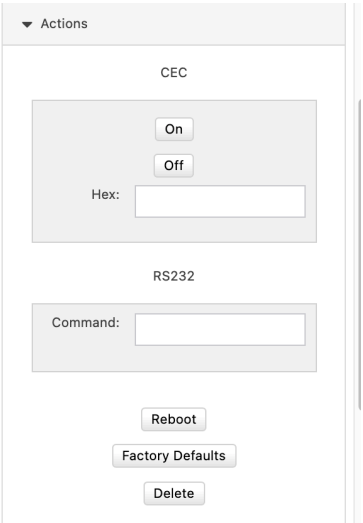


► To delete an encoder or decoder:

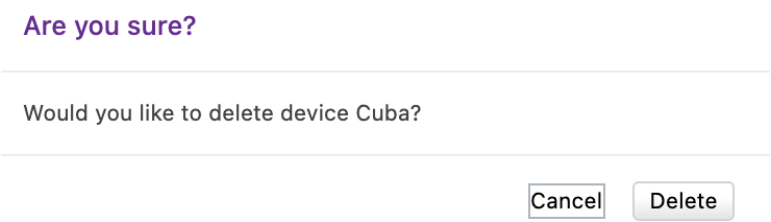
a. Left-click on the desired encoder or decoder name and select Actions from the context menu.



b. Click the Delete button.



3. The following prompt will be displayed when deleting a source.



4. Click the Delete button to confirm the operation. Click the Cancel button to cancel the operation.

5. This same menu can be used to Reboot the device or set the device back to Factory Defaults by clicking the appropriate button.

Sending a CEC Command

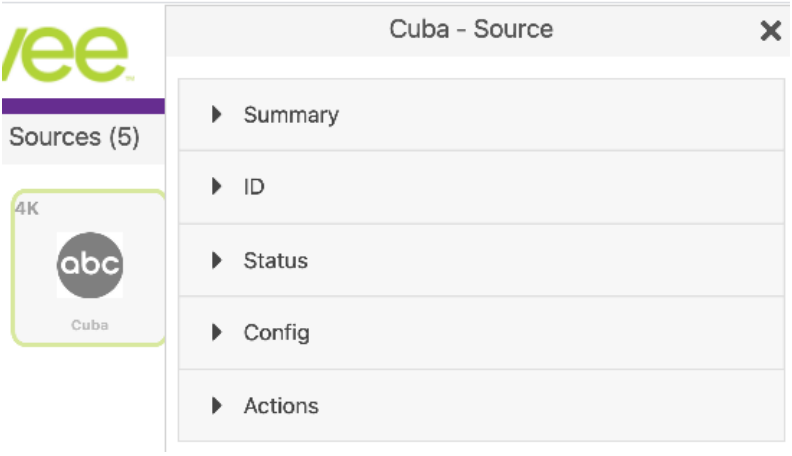
The ZyPer4K, ZyPerUHD60 and ZyPerUHD can send CEC on/off commands from within ZMP. (Note: Decoder only feature for ZyPerUHD/UHD60. Not supported for Encoders) The ZyPer4K and ZyPerUHD60 can send additional CEC hex commands as well.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

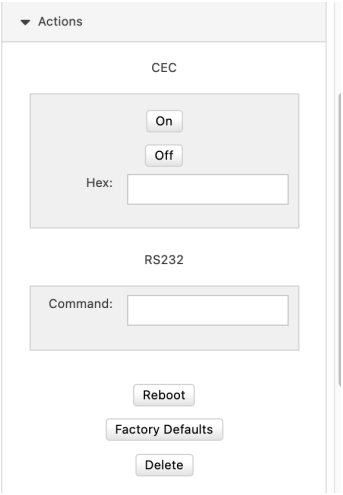
2. Click the Sources tab at the left of the page.

► To send a CEC command:

a. Left-click on the desired encoder or decoder name and select Actions from the context menu. (Note: Decoder only feature for ZyPerUHD/UHD60. Not supported for Encoders.)



b. Click the desired CEC On or OFF button.



3. The ZyPer4K and ZyPerUHD60 can also send Hex commands over CEC. Just type the Hex command into the box. When completed, click anywhere outside of the Hex input box to send the command.

Sending an RS232 String Command

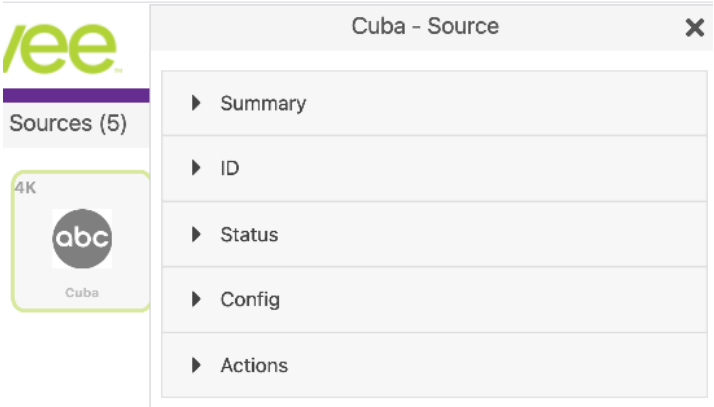
The ZyPer4K, ZyPerUHD60 and ZyPerUHD can be sent RS232 strings from the ZMP to be output on the RS232 port of the unit.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

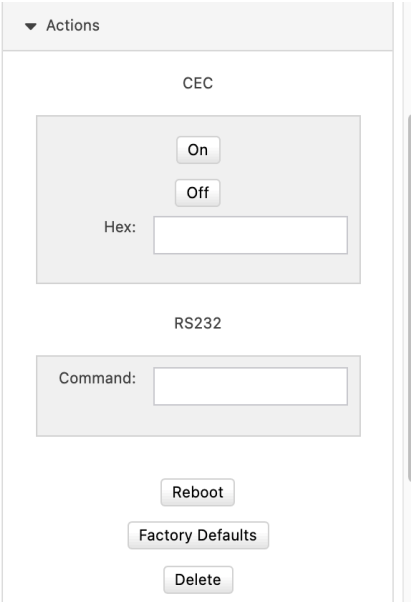
2. Click the Sources tab at the left of the page.

► To send an RS232 string command:

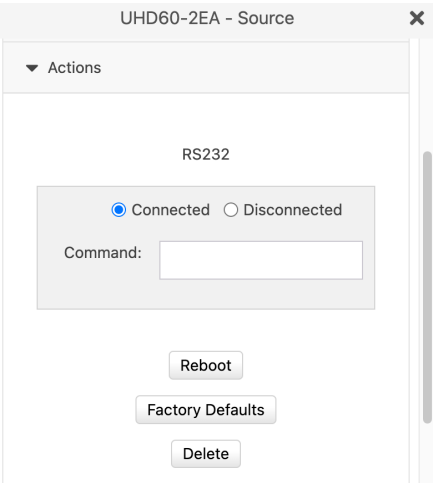
a. Left-click on the desired encoder or decoder name and select Actions from the context menu.



b. Scroll down to the RS232 box and enter desired text. Click anywhere outside the box to send the text string command.



Note: The ZyPerUHD60 has a button to indicate if the encoder/decoder is connected to the ZMP Server for RS232 communication. See the "Switch" command in API Command Listing. Example below is "Connected" Changing this setting will cause the ZyPerUHD60 to reboot.



Updating Firmware

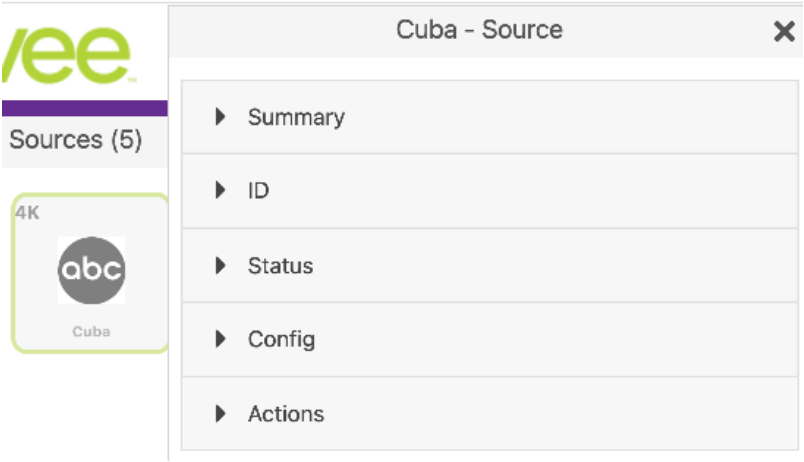
If there is a firmware update available for the ZyPer4K, ZyPerUHD60 or ZyPerUHD, the update can be performed easily from within ZMP.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

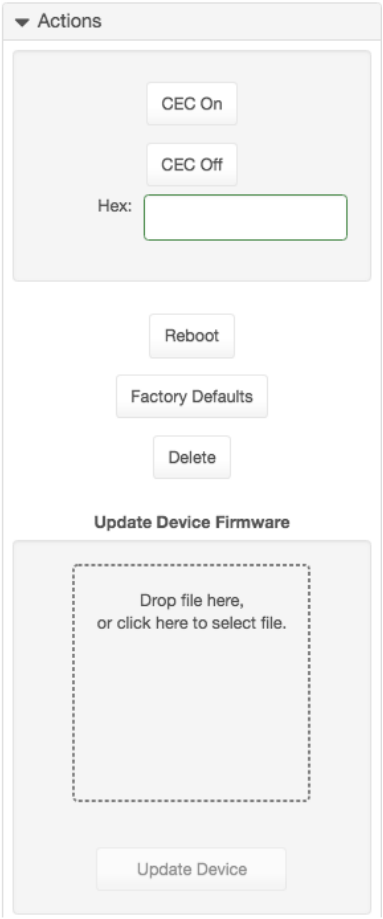
2. Click the Sources tab at the left of the page.

► To update an encoder or decoder:

a. Left-click on the desired encoder or decoder name and select Actions from the context menu.



b. Drag and drop the appropriate firmware update file into the box and click on the "Update Device" button.



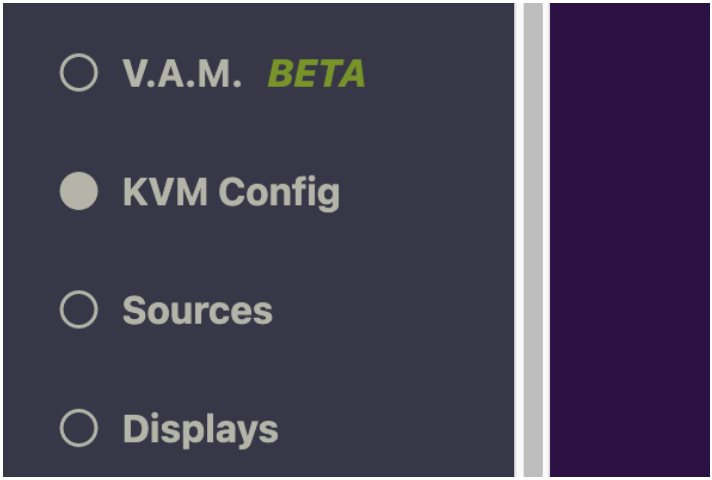
3. You will be prompted to confirm this the desired action. Confirm the action by clicking the Update button. The ZyPer unit will automatically reboot itself once the firmware update is complete.

Configuring KVM Roaming

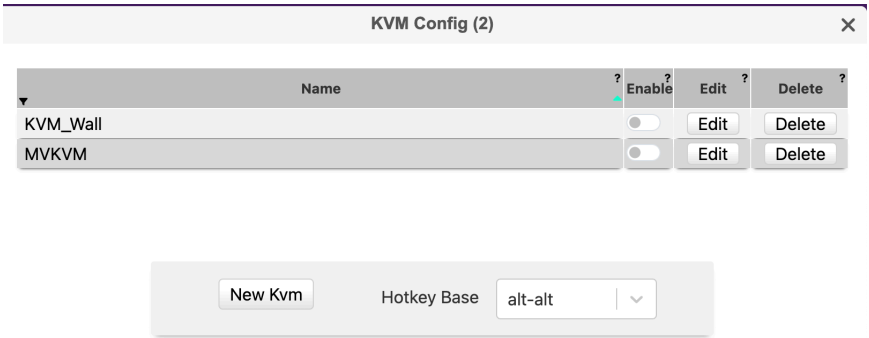
KVM Roaming allows multiple sources keyboard/mouse functionality to be controlled from a single keyboard/mouse connected to a ZyPer decoder. The sources can be shown on individual screens or as part of a multiview (ZyPer4K only) Note: ZyPer4K support is for ZyPer4K-XS/XSE/XR units only. (Black boxes)

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

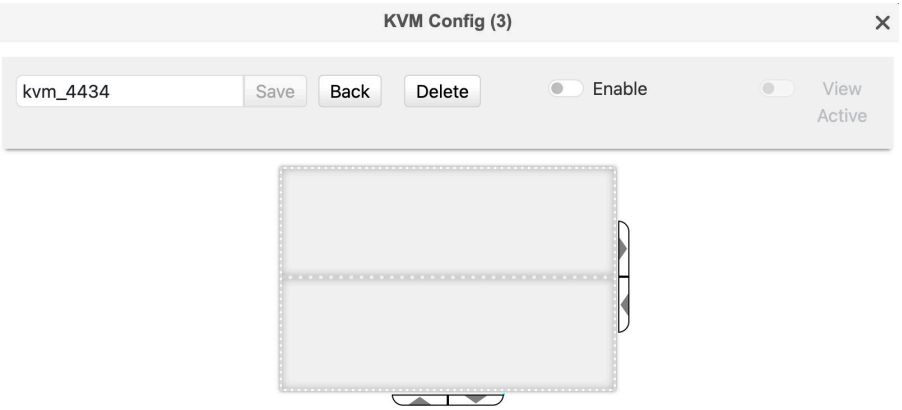
2. Click the KVM Config tab at the left of the page.



3. In the KVM Config pane, click on the "New KVM" button to create a new KVM configuration.

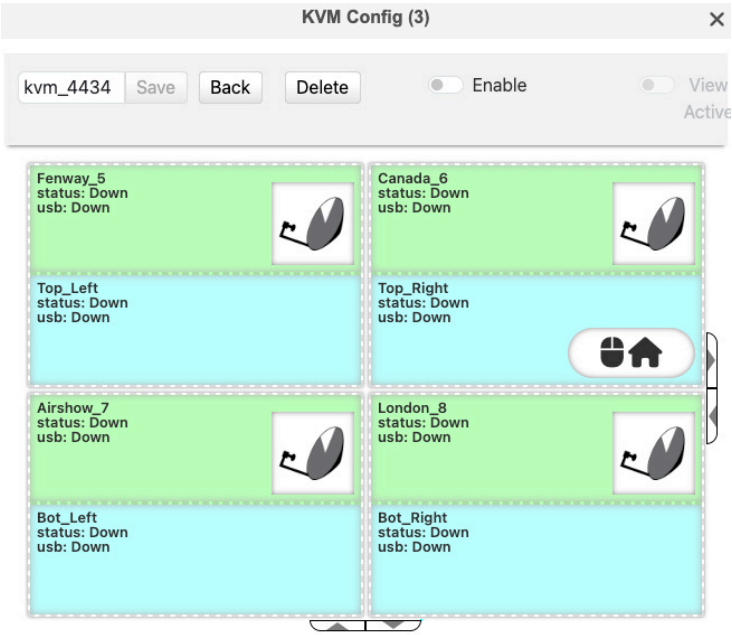


4. This will bring up the KVM Editor.

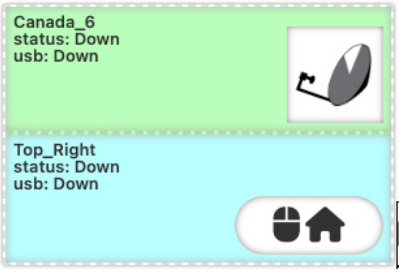


5. Use the up/down or left/right add addtional screens. The graphic above represents a single screen. The top portion is the assigned source (Single source or Multiview). The bottom portion is the assigned display.

6. Drag sources into the top portion of the window and displays into the bottom window. Below is an example of a 2x2 configuration with sources and displays assigned.

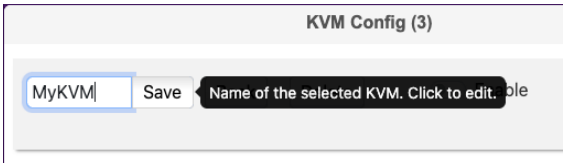


7. Drag the Mouse Icon to the Display/Decoder that actually has the keyboard/mouse connected to it. In this example the keyboard/mouse is connected to the Top_Right decoder.

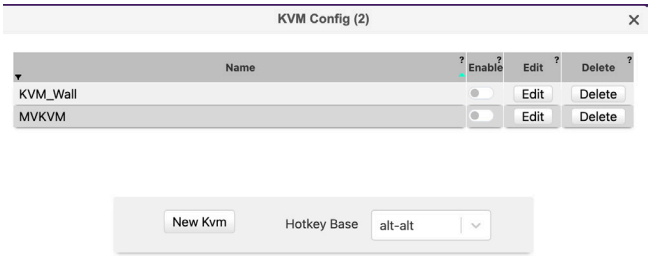


8. Drag the Home Icon to the Display/Decoder that you would like to act as the home display when using the Hot-Key funcitonality. In this example the Home is assigned to the Top_Right decoder.

9. Update the name of your KVM Config in the top left corner and click the "Save" button.



10. Select Hotkey base from the main KVM config pane.



11. To activate a KVM Config, click the "Enable" button for that KVM. Note that multiple KVM can be enabled at the same time, but they cannot share resources. (Encoders or Decoders) Important Note: KVM Config must be disabled to allow normal video routing.

12. Hotkey function will bring selected source to the home screen. Double click Hotkey base. (Alt-Alt for example). Then click

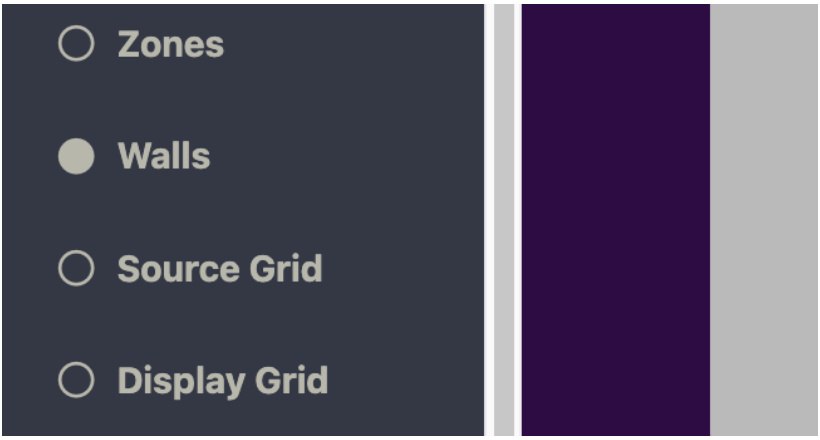
Key: "R" Action: Reset all windows the settings in the saved configuration
Key: "S" Action: Swap the window the mouse is in with the home position

Creating Video Walls

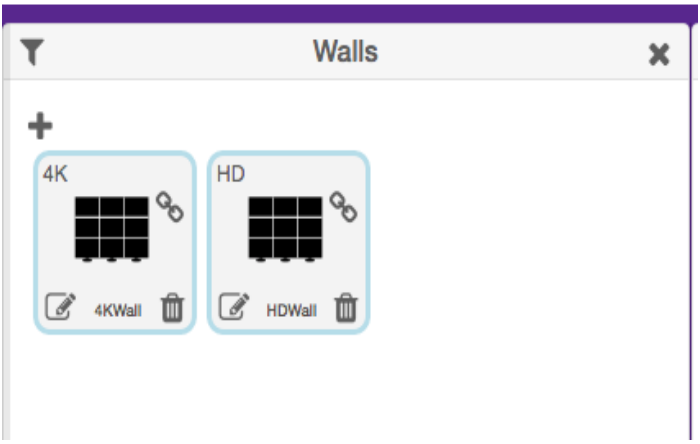
One of the purposes of the Management Platform is to create and manage video walls. A video wall is a collection of displays or projectors arranged in a square or rectangular fashion. The source is then "mapped" to each display, thereby creating one large display from multiple, smaller displays.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

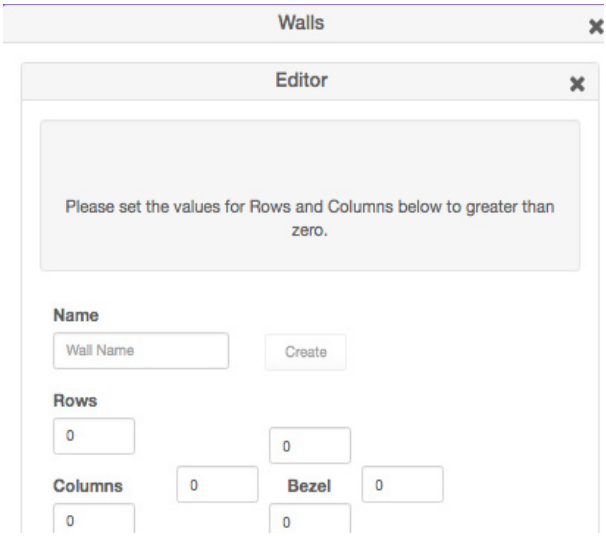
2. Click the Walls tab at the left of the page.



3. In the Walls pane, click on the + button to create a new video wall.



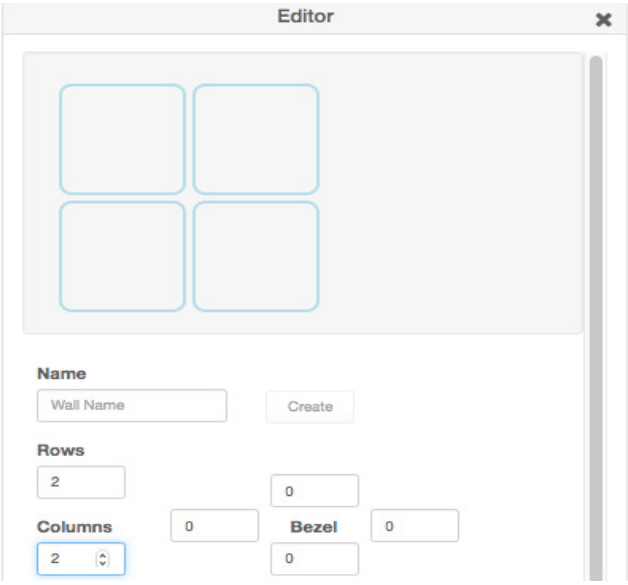
4. This will bring up the wall Editor.



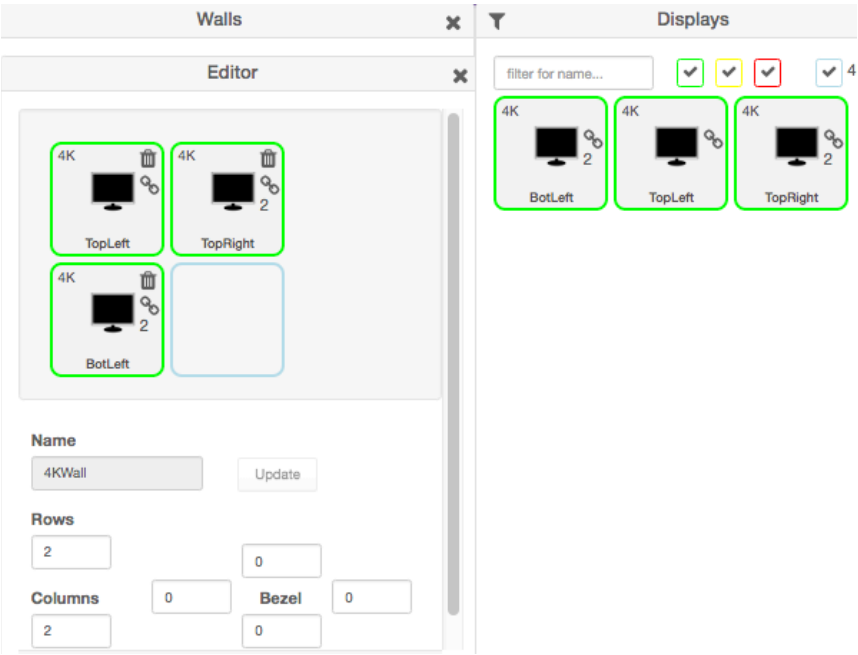
5. Enter the number of display columns in the Number of Columns field.

6. Enter the number of display rows in the Number of Rows field.

   In this example, we have arranged our displays in two rows and two columns. This is a blank 2x2 video wall:



7. Drag Displays from the Display pane into the Editor to populate the wall. Provide a name for the new video wall in the Name field. Values in this field cannot contain spaces. Since more than one video wall can be created, always be sure to provide a descriptive name. This field cannot be blank.

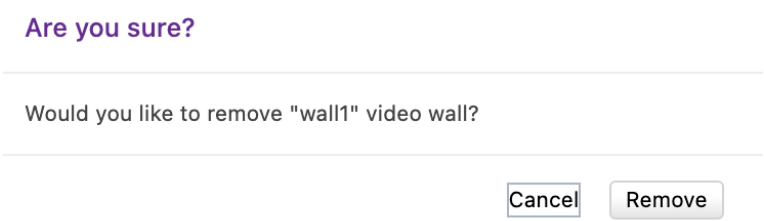


8. Leave the Bezel Top, Bezel Bottom, Bezel Left, and Bezel Right set to 0. Bezel compensation will be covered in the next section.

9. Click the Create button.

10. The new Wall will now be available within the Walls window.

11. Video Walls are deleted by clicking on the small Trash Can icon. The user will be prompted to confirm deletion.



Bezel Compensation

Every video output device has an area where video is not displayed. This area is called the bezel. Bezel compensation takes this area into account when a single video source is divided and displayed on multiple output devices.

1. Check the output on the video wall and identify any misaligned edges. For best results, it is recommended to use a static video pattern for this test.

   In the illustration on the left, we have a 2x2 video wall without bezel compensation. Note the ZeeVee logo is not aligned correctly across all four displays. On the right, bezel compensation is used to fix the issue.

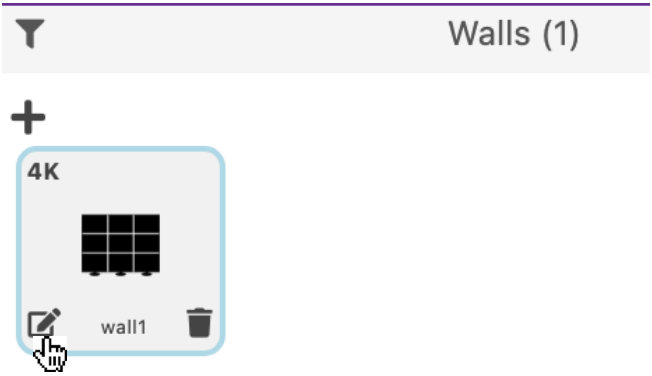
Without Bezel Compensation

With Bezel Compensation

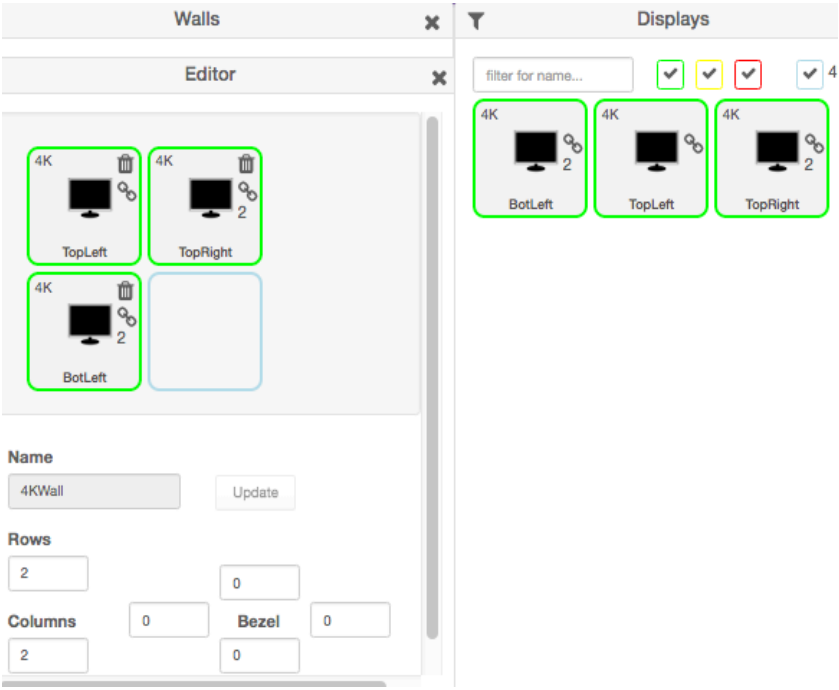
   It is recommended that when the video wall is set up for the first time, that these values be set to zero. Bezel compensation can be changed at any time.

   Bezel compensation is always measured in pixels.

2. Left click the edit video wall button in the lower left corner of the desired wall.



3. The video wall editor will come back up onto the screen.



4. Enter the desired values, in pixels, for each bezel field: Bezel Top, Bezel Bottom, Bezel Left, Bezel Right.

5. Once the desired values have been entered, click the Update button. This will save the new settings.

6. Check the picture on the displays. Repeat steps 2 - 5 as necessary.

7. Note: Only the ZyPer4K and ZyPerUHD60 allow bezel adjustment. Video walls with the ZyPerUHD do not allow bezel adjustment.

8. Note: If the PLUS sign or trash can for multiviews or video walls is not visible in your browser; the display may have entered into tablet mode. You can force the system back to "Desktop" mode with the following command line addition in the browser. Add /?destop=true. Example: http://172.16.6.111/maestroz/app/?desktop=true

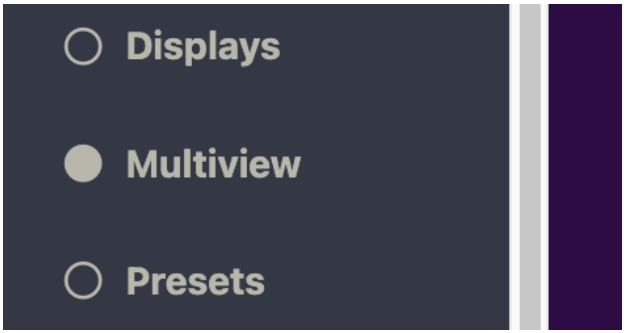
Creating a Multiview Screen

One of the purposes of the Management Platform is to create and manage multiview screens. A multiview screen is a collection of sources arranged on a single display. There can be up to 19 sources displayed on a single display in a variety of preset patterns.

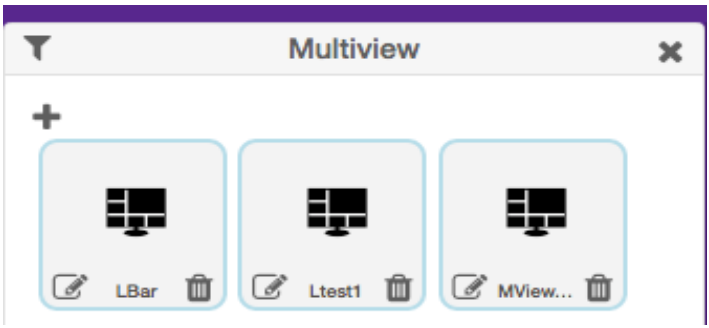
Note: Only the ZyPer4K family supports multiview screens. This feature is not supported by the ZyPerUHD or ZyPerUHD60.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

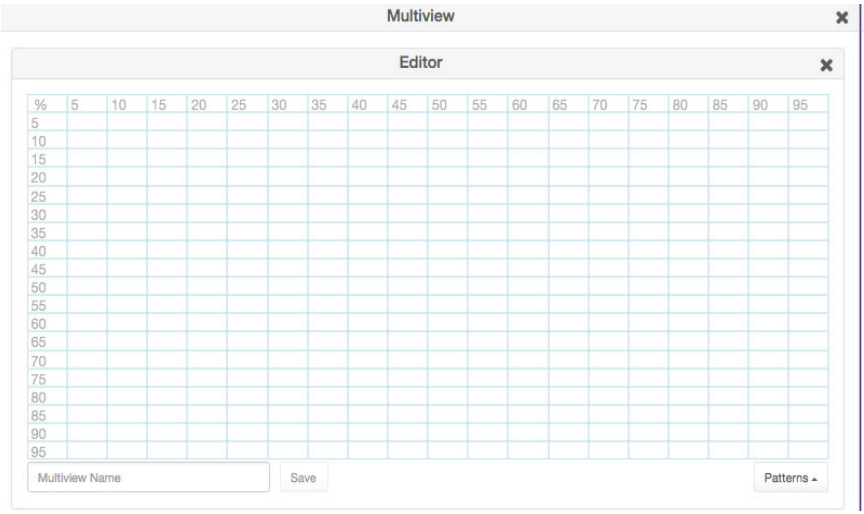
2. Click the Multiview tab at the left of the page.



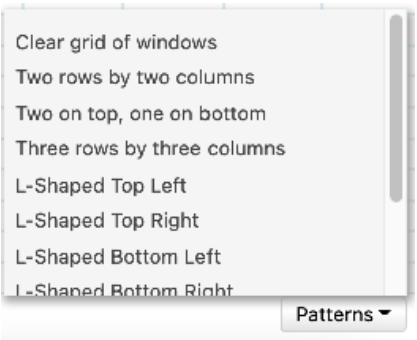
3. In the Multiview pane, click on the + button to create a new video wall.

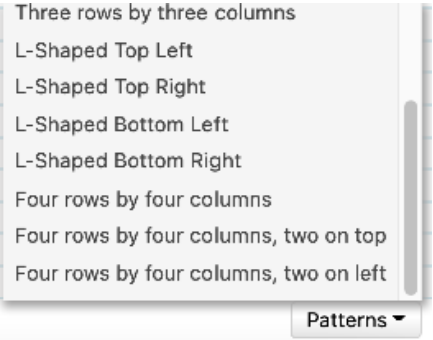


4. This will bring up the Multiview editor where you can create and customize a multiview display.

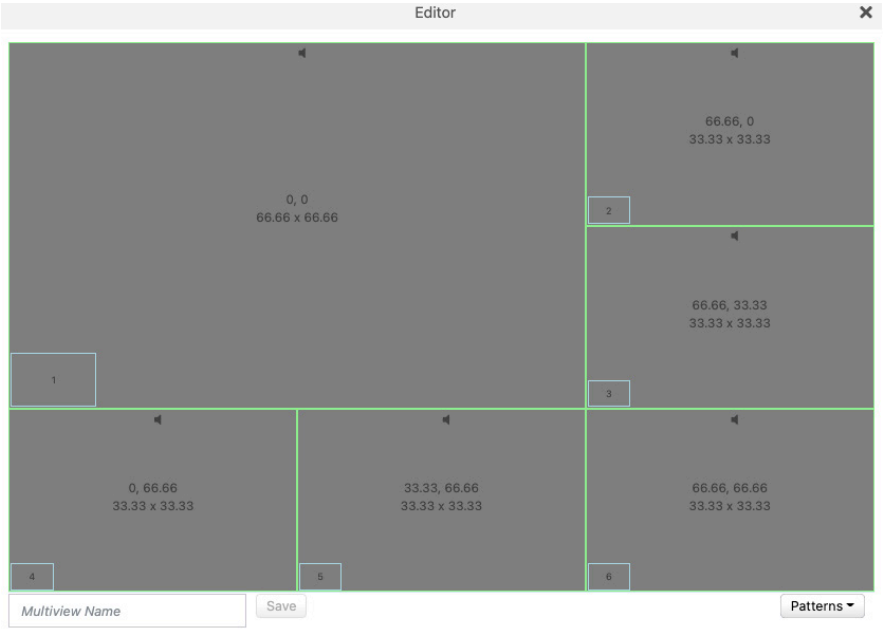


5. Click on the Patterns button in the lower right corner to bring up a list of pre-defined patterns.

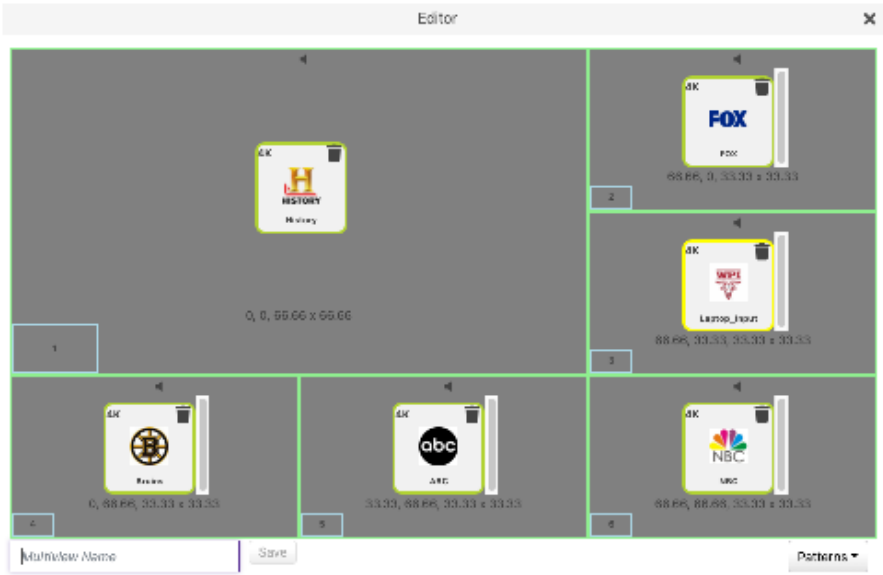




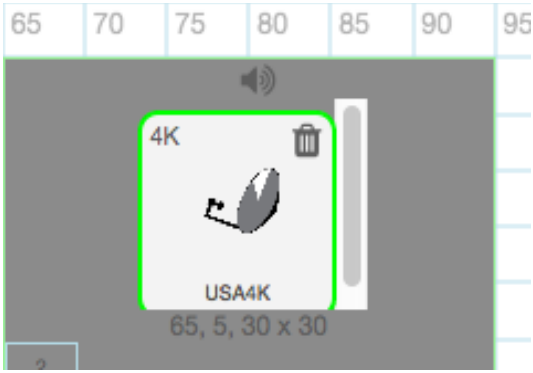
6. After selecting a pattern you will see the grid fill in with the places to drag sources.



7. Drag sources into the various locations. Note: The same source can be dragged into multiple windows as long as the windows are the same size. Dragging a source onto different size windows is NOT supported.



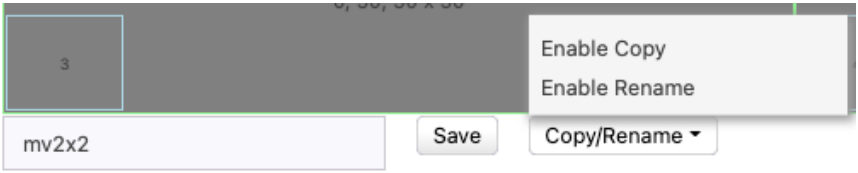
8. If desired, assign Audio from one of the Sources. This is done by clicking on one of the small Audio symbols in the top center of each Window. Note that only audio from one source is supported.



9. Give the Multiview a name and press the Save button.

10. To use the Multiview, drag the newly created multiview onto a Display in the Display pane.

You can rename or create a copy of a multiview by going back into any existing multiview and clicking the Copy/Rename button next to the Save button.



Click "Enable Copy", enter the new multiview name and click Save.

Click "Enable Rename", enter the new name for existing multiview and click Save.

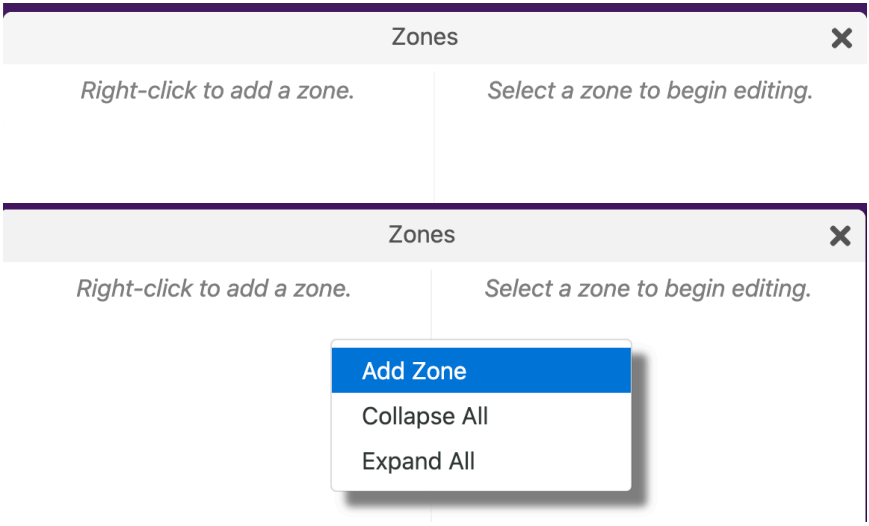
Creating Zones

In many installations, displays are placed in more than one room. These rooms are often referred to as zones. Creating a Zone, using the Management Platform, allows you to organize these displays in a group. Video Walls can also be added to Zones.

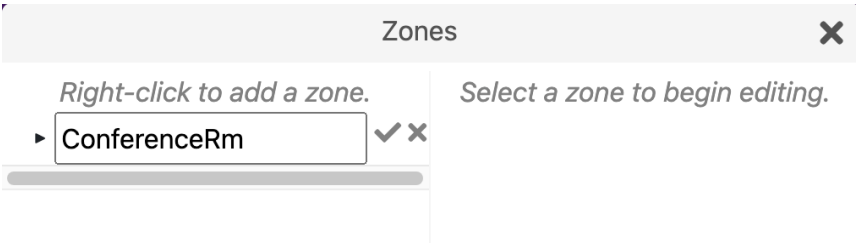
1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Zones option at the left of the page.

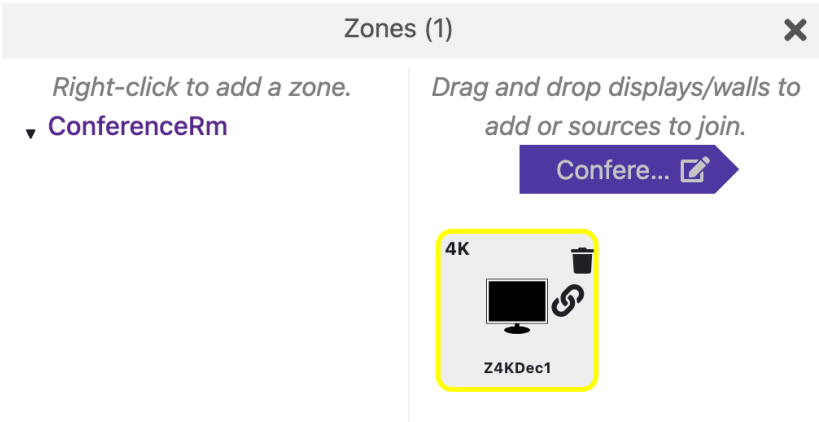
3. In the Zones window, right click to add a Zone.



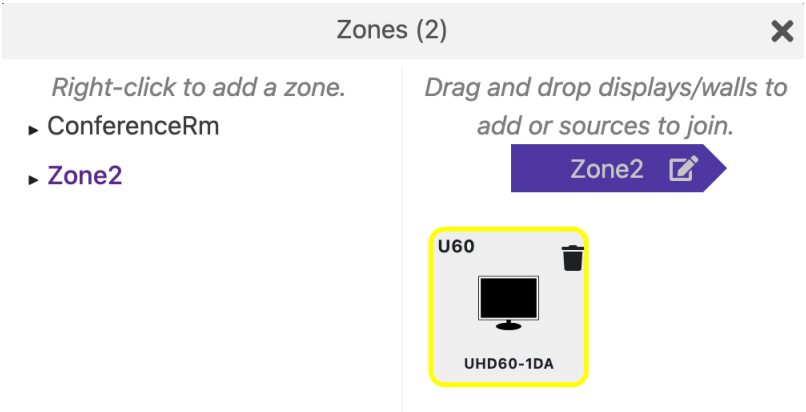
4. Provide a name for the Zone. If a custom name is used, it must not contain spaces. In this example, we will call our Zone, "ConferenceRm", since our displays are installed in a conference room. It is recommended that a unique and descriptive name be used to identify each Zone.



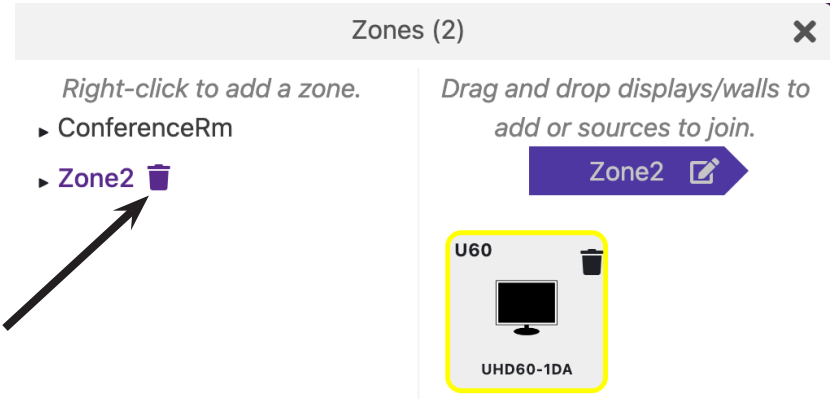
5. Select the Zone and then you can drag/drop Displays into the Zone from the Displays Panel.



6. Additional Zones can be added as desired by following the same process.



7. Zones are deleted by clicking on the small Trash Can icon. The user will be prompted to confirm deletion.

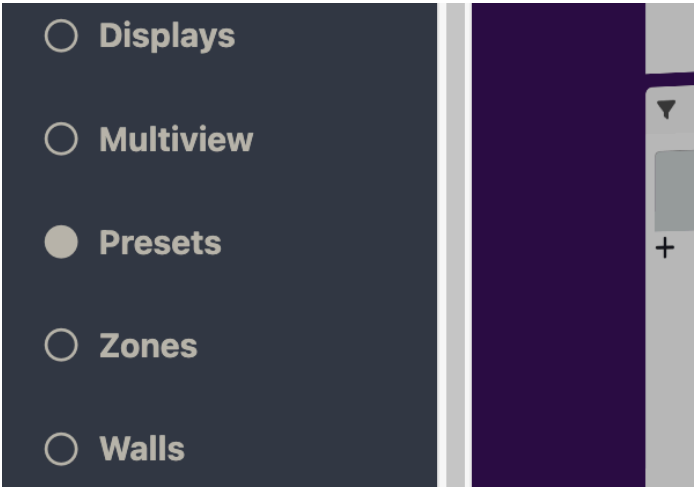


Creating Presets and Schedules

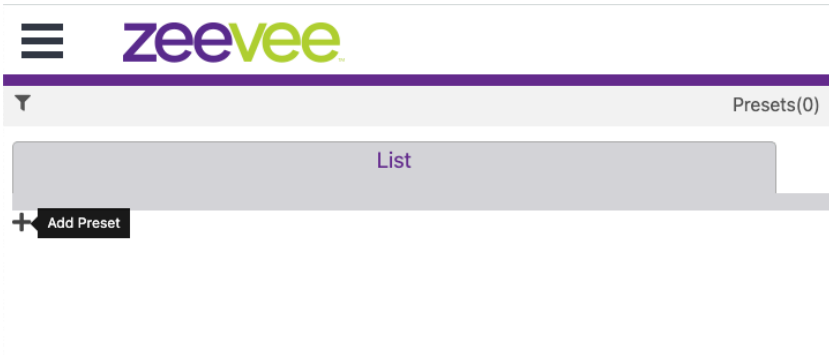
Presets are snapshots of a system configuration at a given time. You can save presets to the system and return to these configurations manually or via the scheduling function.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

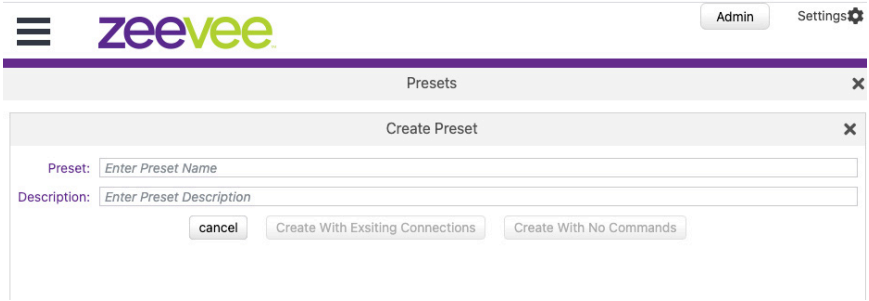
2. Click the Presets option at the left of the page.



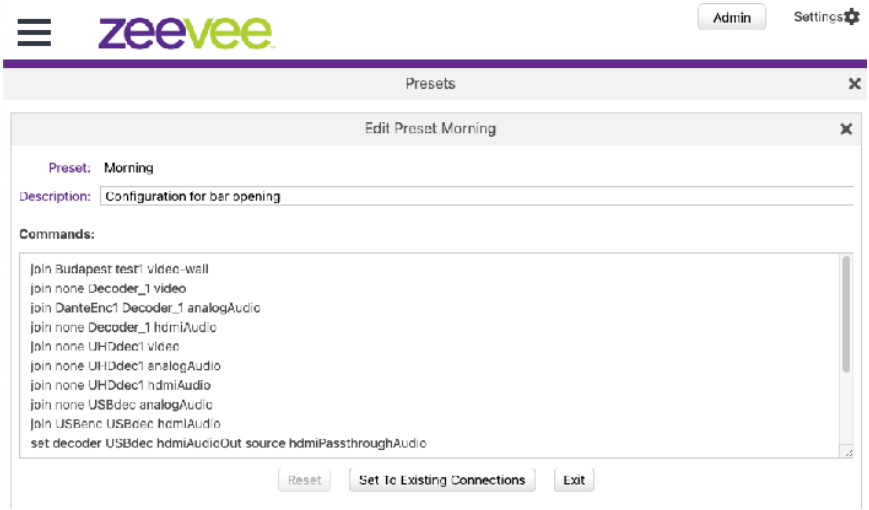
3. In the Presets/List window, click on the + symbol to add a Preset.



4. This will open the Create Preset window. Enter a name and description for this preset.

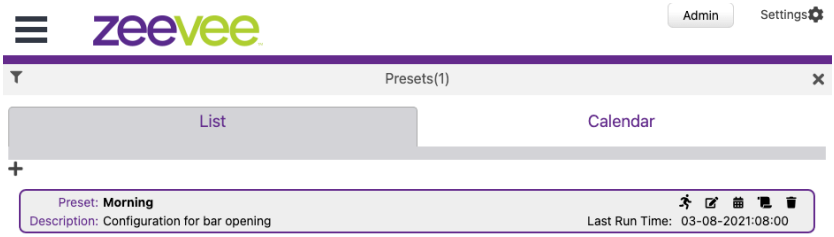


5. You can now Create the preset using the current state of the system. "Create with Existing Connections" or you can "Create with No Commands" and add the API commands manually.

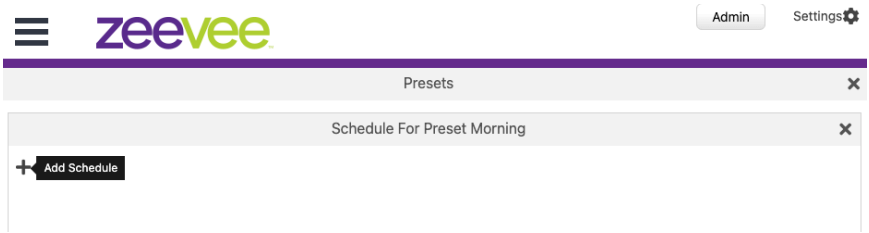


6. The example above is "Create with Existing Connections". Notice the relevant API commands to return to the current system state are listed under Commands: You can manually edit this list to add/remove/change commands as desired. Hit the Exit key when done creating the preset.

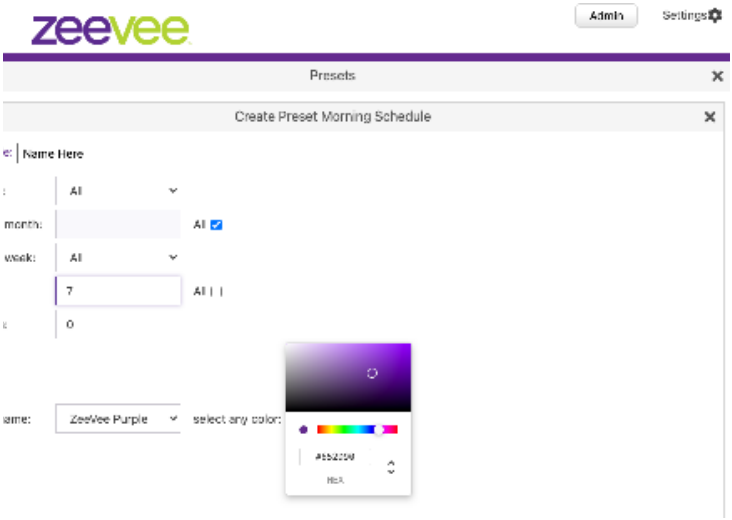
7. The created preset is now available. Options include, running the preset manually, editing the preset, scheduling the preset, showing a log of when preset was run and deleting the preset.



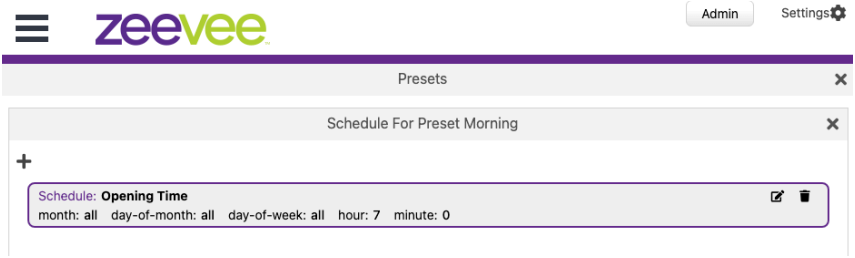
8. To add the preset to the schedule/calendar click the small calendar icon. (3rd icon in). This will open the window shown below. Click the + symbol to "Add Schedule".



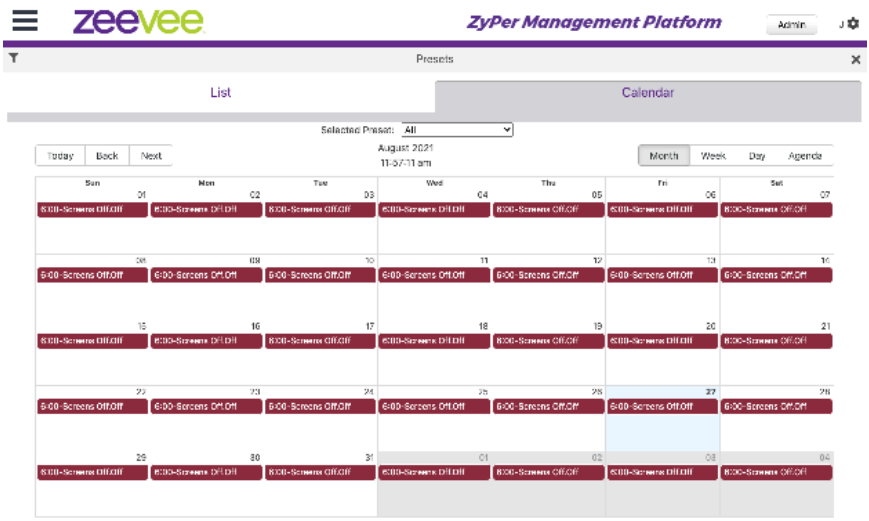
9. Assign a name to the schedule here. It can be the same as the preset name or something different. Select the date of the month - enter value 1 - 31. Please a checkmark next to All for everyday of the month. Day of the Week - Select day to run Preset by day of week, weekend, weekday, or all. Weekends are Saturday and Sunday, and Weekdays are Monday - Friday. Hour - Values range 0 - 23. Minute - values range 0 - 59. Select color for the preset to appear in the calendar page. Click Create when done.



10. In the example below the name of the Schedule is "Opening Time" and has been set to occur every day at 7am. Click the X next to "Schedule For Preset Morning" to exit this page.



11. This preset will now appear in the Calendar page.



12. You can add as many preset configurations to the calendar as you like. You also run the same preset at different days/times as desired.

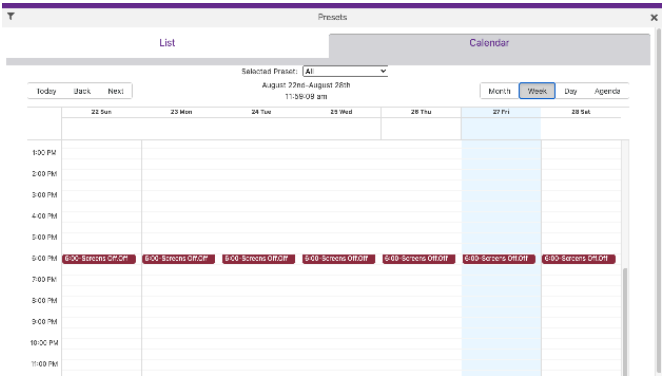
Note: The following API commands are used to manually configure NTP servers or set the time for customers without Internet access or a local time server:

set server date ntpServer address <domainName>

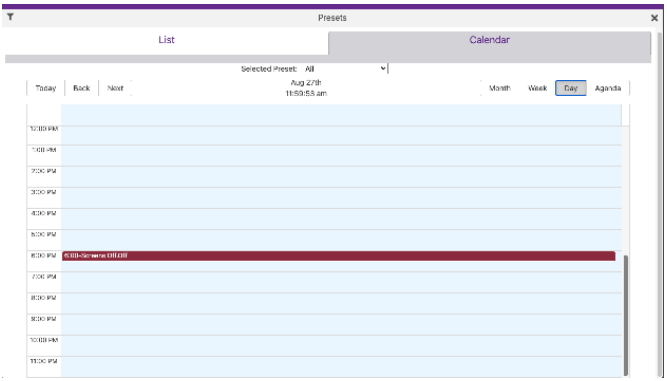
set server date manual month <int> day <int> year <int> hour <int> minute <int>

13. The Calendar can be viewed in Month, Week, Day or Agenda mode.

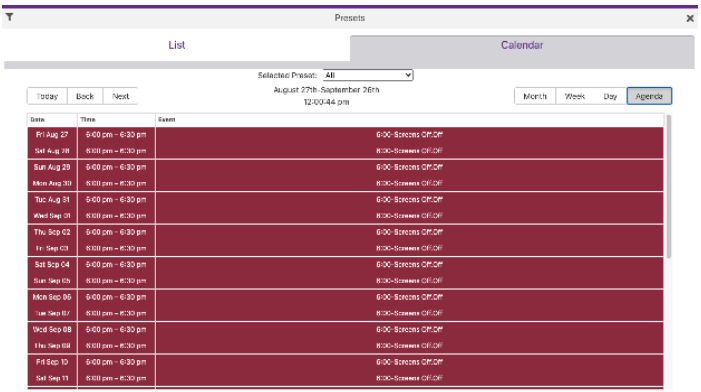
Week Mode shown below:



Day Mode shown below:



Agenda Mode shown below:



Source Grid Page

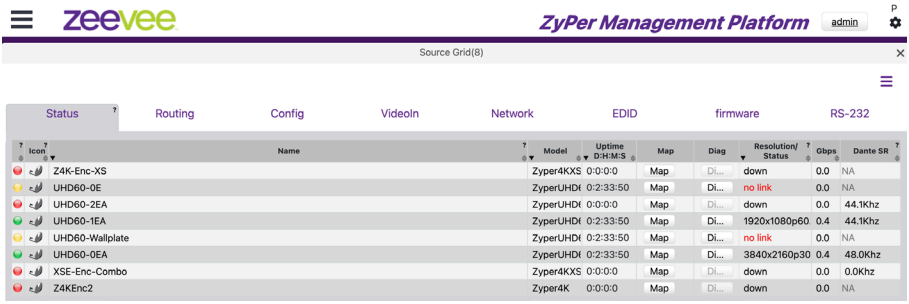
This page is used to show all encoders in a text based grid. Information shown includes device name, type, MAC address, Model, IP address and EDID. The user can also export the entire contents of the Source Grid to a .CSV file.

Source Grid

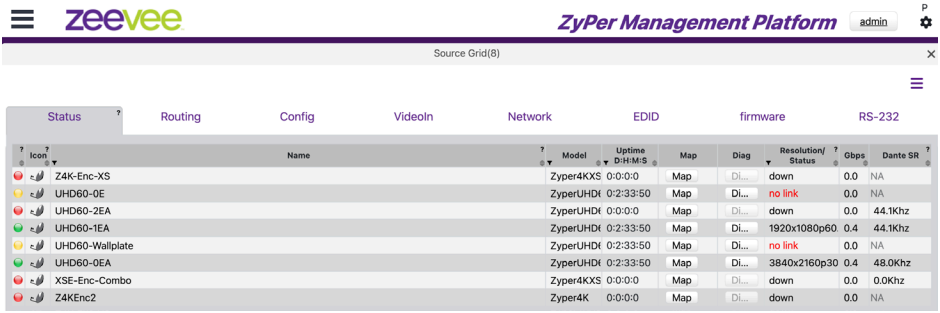
1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Source Grid option at the left of the page.

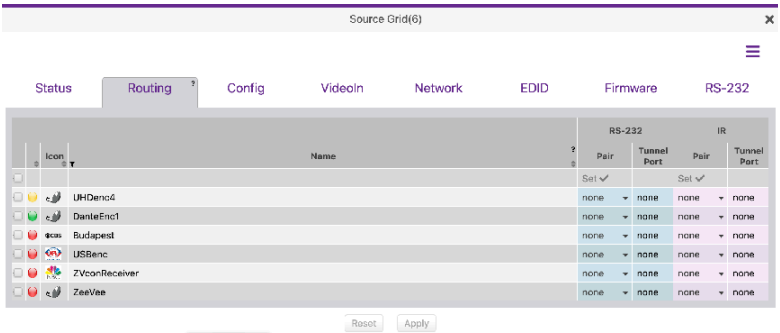
3. Source Device/Encoder information will be shown in a grid format. There are independent tabs for Status, Routing, Config, VidoeIn, Network, EDID, Firmware and RS-232. You may want to close other windows to be able to maximize the size of the grid.



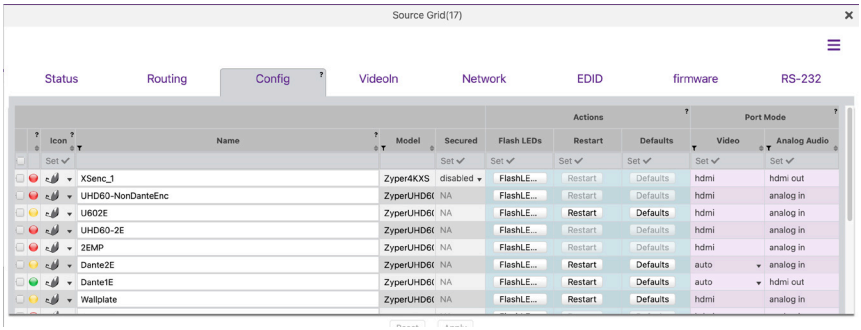
4. The Status tab shows basic information and allows a quick diagnostic by clicking the Diag button. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns. (Note: Dante SR is input audio frequency)



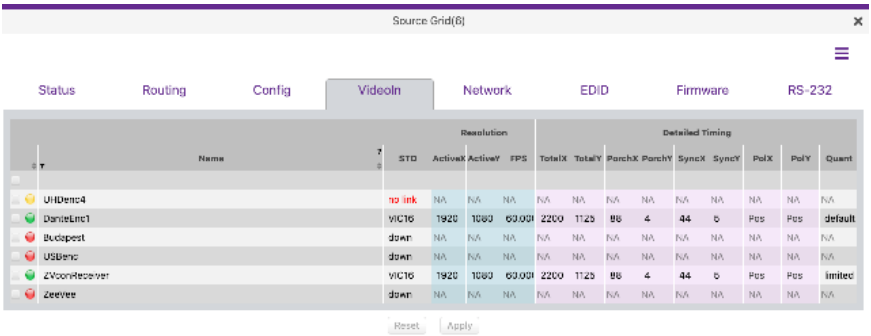
5. The Routing tab allows you to manually route RS-232 and IR signals. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.



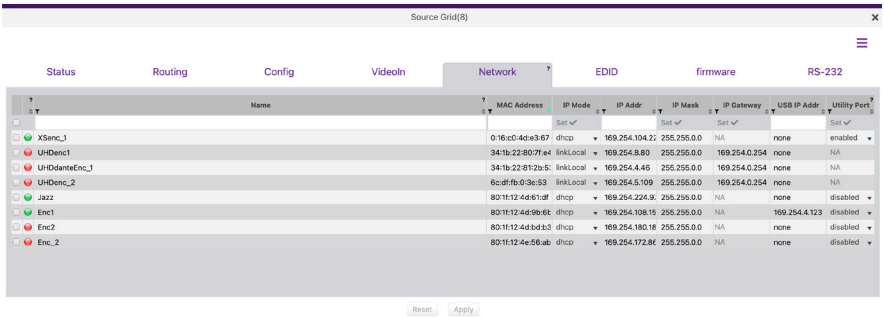
6. The Config tab will let you change encoder names and manually restart or factory default the encoders. FlashLEDs will cause both Video and Power LEDs on the Z4K encoder to flash for a few seconds. (Status light on ZyPerUHD60). The user can also change Icons associated with the encoder; including the ability to upload custom Icons. (JPG and PNG formats are both acceptable) Click the small arrow next to the Icon to bring up a menu of available options. Z4K-XS/XSE units will show if the Secured Communication feature is enabled or disabled. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.



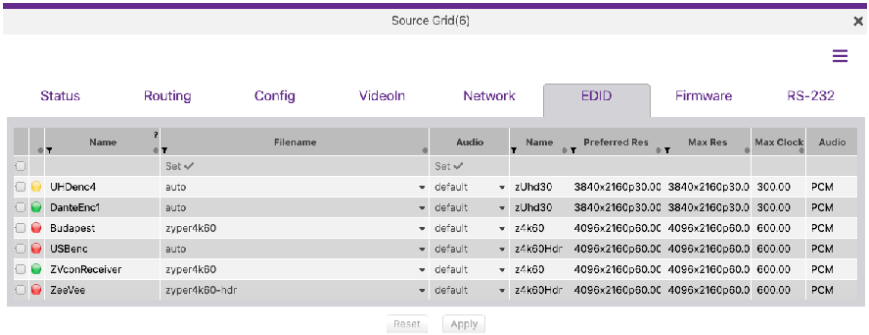
7. The VideoIn tab shows information about connected source Color Space and Resolution. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

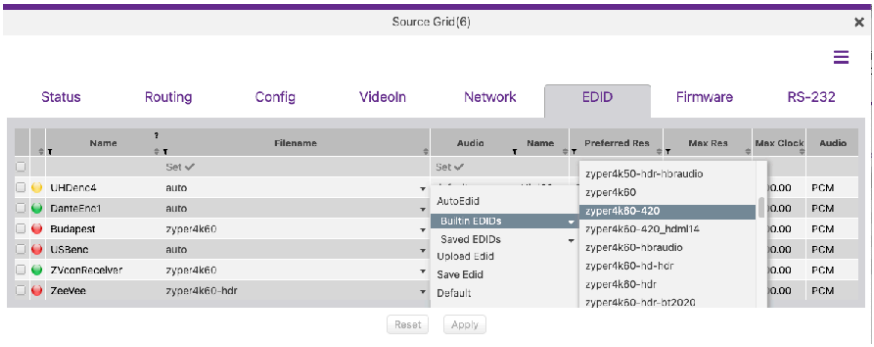


8. The Network tab has details such as IP address and Subnet Mask. IP Address of any integrated Icron USB devices will also be shown. The user can also enable or disable the 1Gb Utility Port on the ZyPer4K. Utility Port setting on Dante Enabled Dual Port ZyPerUHD60 encoders can enable/disable Dante to that port. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

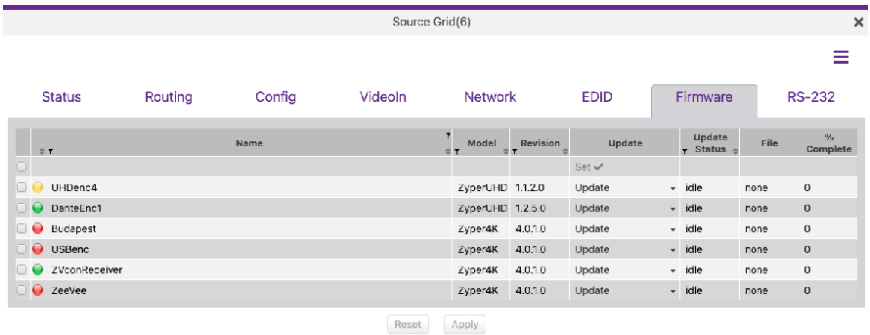


9. The EDID tab will show current EDID information allow the user to manually assign an EDID to the encoder if desired. Click the small arrow in the Filename column for a specific encoder to change the EDID. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

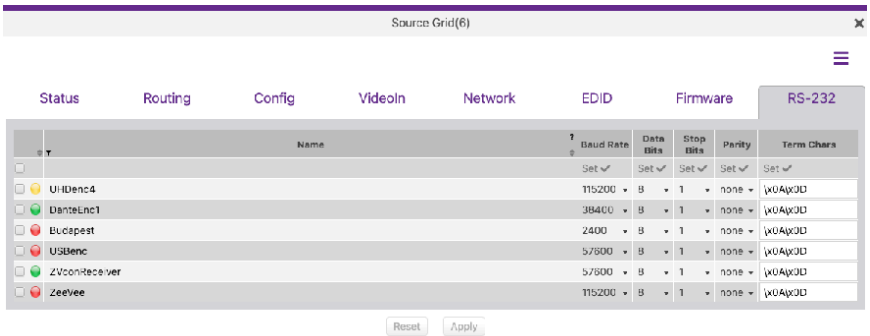




10. The Firmware tab will show the current version of firmware installed on the encoders and allow you to manually update to another version if needed. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

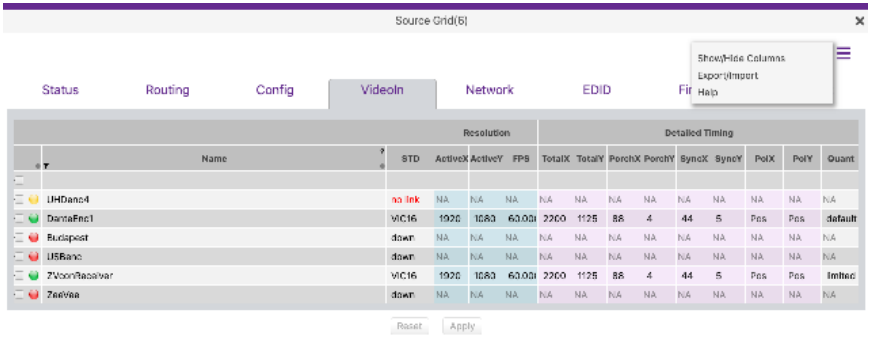


11. The RS-232 tab will show current RS-232 configuration settings and allow you to make changes. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

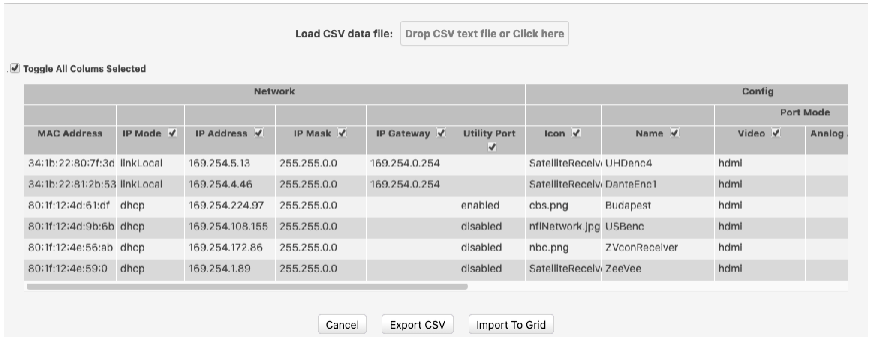


12. Exporting the Source Grid to a CSV file

Clicking on the 3 parallel bars from any tab in the Source Grid will allow you to save the Source Grid contents to a CSV file.



Select the Export/Import option.



Note that there is a check box in the upper left allowing you to select all columns to be exported. You can also individually select what columns you wish to export.

Once you click the Export CSV button, the file will be downloaded to your computer. The CSV file can be opened and viewed by many applications including Microsoft Excel.

The CSV can be edited and uploaded back into the system if desired. Using the process above, drag the edited CSV file into the "Load CSV data file" box and click the Import to Grid button.

Display Grid Page

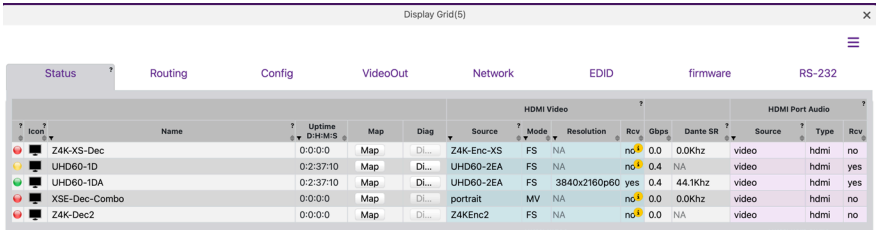
This page is used to show all decoders in a text based grid. Information shown includes device name, type, MAC address, Model, IP address, EDID and Connected Encoder. The user can also export the entire contents of the Display Grid to a .CSV file.

Display Grid

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

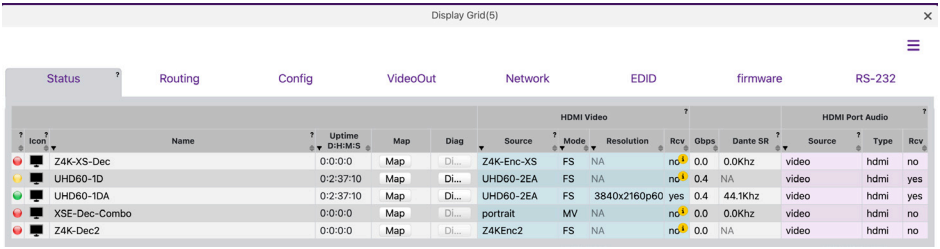
2. Click the Display Grid option at the left of the page.

3. Display Device/Decoder information will be shown in a grid format. There are independent tabs for Status, Routing, Config, VidoeOut, Network, EDID, Firmware and RS-232. You may want to close other windows to be able to maximize the size of the grid. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.



					HDMI Video						HDMI Port Audio		
Icon	Name	Uptime D:H:M:S	Map	Diag	Source	Mode	Resolution	Rcv	Gbps	Dante SR	Source	Type	Rcv
	Z4K-XS-Dec	0:0:0:0	Map	Di...	Z4K-Enc-XS	FS	NA	no	0.0	0.0Khz	video	hdmi	no
	UHD60-1D	0:2:37:10	Map	Di...	UHD60-2EA	FS	NA	no	0.4	NA	video	hdmi	yes
	UHD60-1DA	0:2:37:10	Map	Di...	UHD60-2EA	FS	3840x2160p60	yes	0.4	44.1Khz	video	hdmi	yes
	XSE-Dec-Combo	0:0:0:0	Map	Di...	portrait	MV	NA	no	0.0	0.0Khz	video	hdmi	no
	Z4K-Dec2	0:0:0:0	Map	Di...	Z4KEnc2	FS	NA	no	0.0	NA	video	hdmi	no

4. The Status tab shows basic information and allows a quick diagnostic by clicking the Diag button. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.



					HDMI Video						HDMI Port Audio		
Icon	Name	Uptime D:H:M:S	Map	Diag	Source	Mode	Resolution	Rcv	Gbps	Dante SR	Source	Type	Rcv
	Z4K-XS-Dec	0:0:0:0	Map	Di...	Z4K-Enc-XS	FS	NA	no	0.0	0.0Khz	video	hdmi	no
	UHD60-1D	0:2:37:10	Map	Di...	UHD60-2EA	FS	NA	no	0.4	NA	video	hdmi	yes
	UHD60-1DA	0:2:37:10	Map	Di...	UHD60-2EA	FS	3840x2160p60	yes	0.4	44.1Khz	video	hdmi	yes
	XSE-Dec-Combo	0:0:0:0	Map	Di...	portrait	MV	NA	no	0.0	0.0Khz	video	hdmi	no
	Z4K-Dec2	0:0:0:0	Map	Di...	Z4KEnc2	FS	NA	no	0.0	NA	video	hdmi	no

Status Icon Color

Every Display features a status indicator, the color of which corresponds to the device's current state.

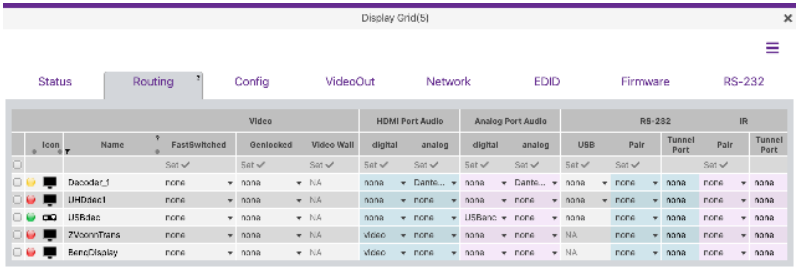
Green - Connected to the network and an active video source is detected.

Yellow - Connected to the network and no HDMI connection to the monitor.

Orange - Connected to the network, HDMI connection to monitor detected but no active video join established.

Red - Not connected to the network or the ZyPer device is powered off.

5. The Routing tab allows you to manually route Video, Video Wall, USB, RS-232 and IR signals. Make changes by clicking into the small down arrow. Field will turn green indicating an upcoming change. Press the Apply button to implement the change. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.



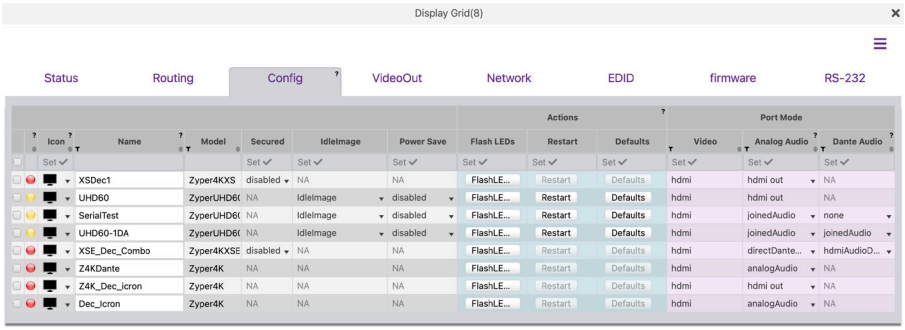
6. The Config tab will let you change decoder names and manually restart or factory default the decoders. It also lets you know what video port is active if more than one port is available and if the Analog audio output is being used.

   FlashLEDs will cause both Video and Power LEDs on the Z4K decoder to flash for a few seconds. (Status light on ZyPerUHD60).

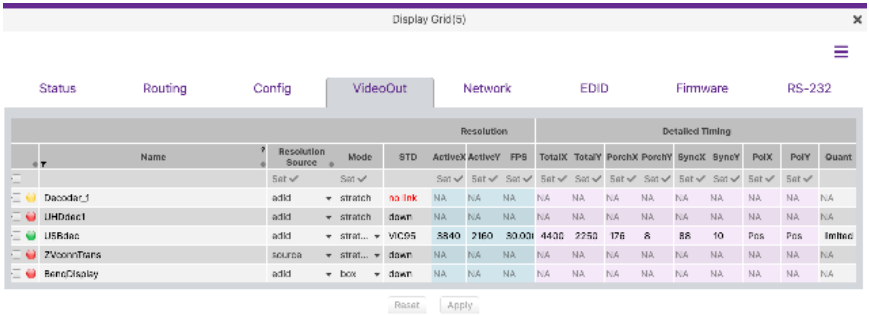
   The user can also change Icons associated with the encoder; including the ability to upload custom Icons. (JPG and PNG formats are both acceptable) Click the small arrow next to the Icon to bring up a menu of available options.

   You can also change the ZyPerUHD or ZyPerUHD60 default "No Source Found" background screen. (IdleImage) Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

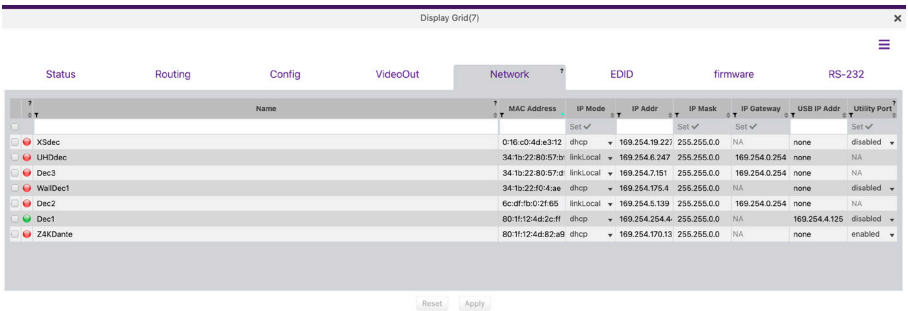
Note: ZyPerUHD IdleImage must be 1280x720 in size and will output from decoder at that resolution. ZyPerUHD60 IdleImage must be 1920x1080 in size and will output from decoder at that resolution.



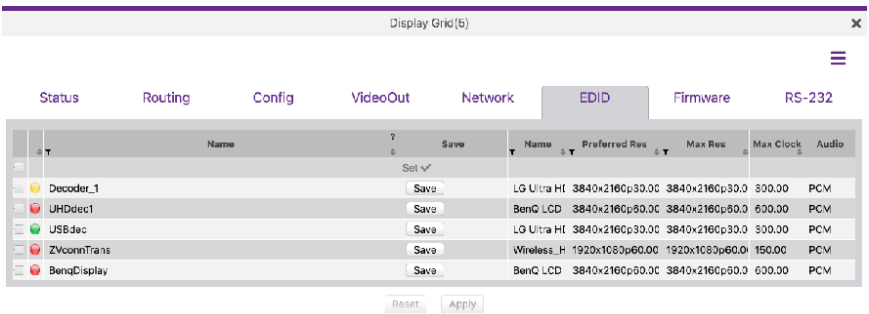
7. The VideoOut tab shows information about the resolution source, scaling mode, color space and active resolution. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

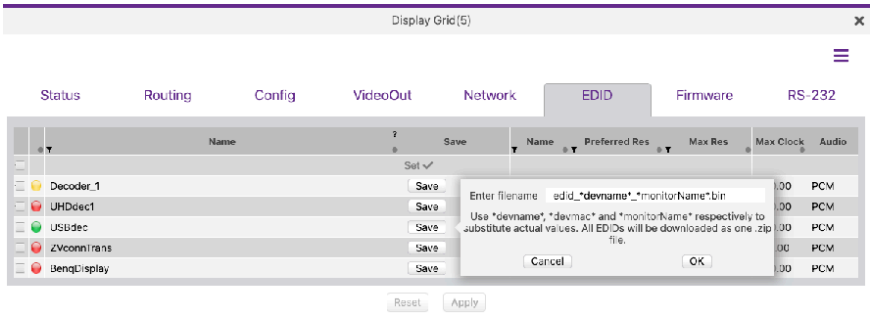


8. The Network tab has details such as IP address and Subnet Mask. IP Address of any integrated Icron USB devices will also be shown. The user can also enable or disable the 1Gb Utility Port on the ZyPer4K. Utility Port setting on Dante Enabled Dual Port ZyPerUHD60 decoders can enable/disable Dante to that port. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

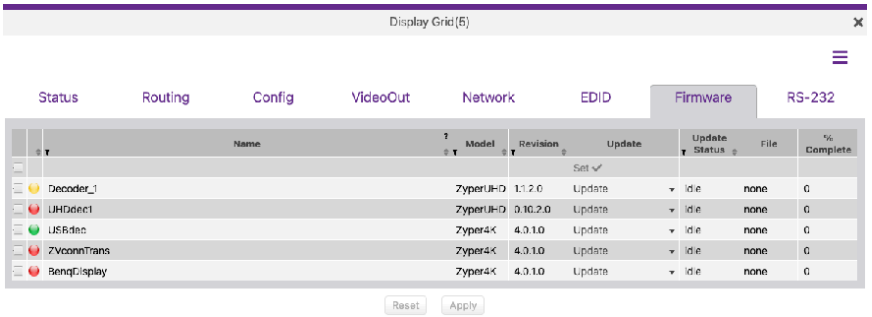


9. The EDID tab will show current EDID information for the connected display and allows the save the EDID to a file. Click the Save button to save the EDID. Note that both a text and binary version will be saved and available to the user. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

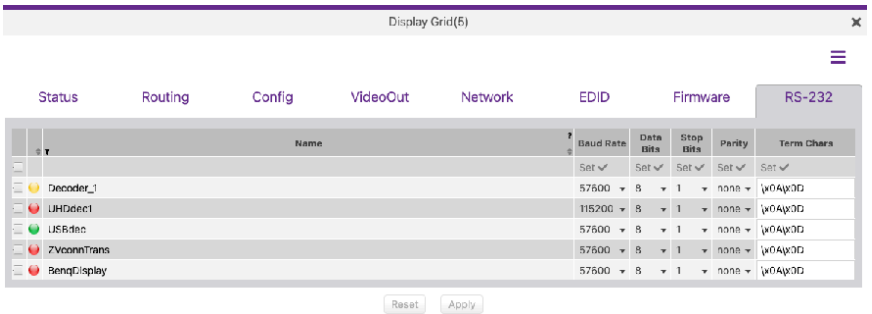




10. The Firmware tab will show the current version of firmware installed on the decoders and allow you to manually update to another version if needed. Note that clicking the 3 horizonatal lines in the top right will allow you to show or hide columns.

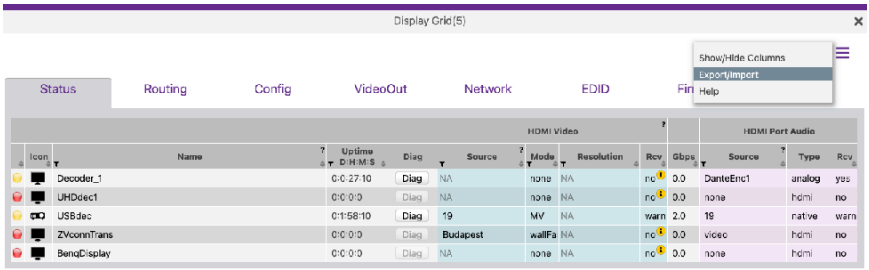


11. The RS-232 tab will show current RS-232 configuration settings and allow you to make changes. Note that clicking the 3 horizonatal lines in the top right will allow you to show or hide columns.

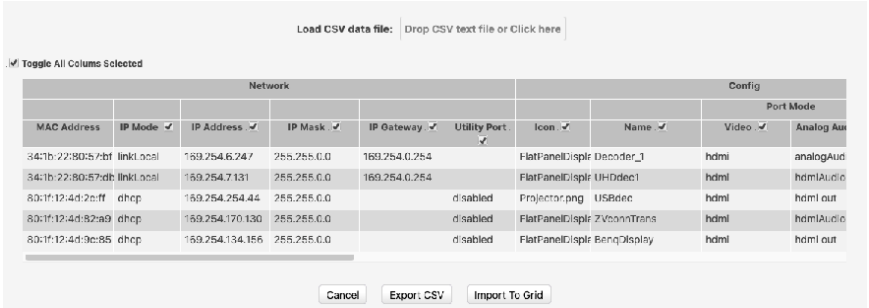


12. Exporting the Display Grid to a CSV file

Clicking on the 3 parallel bars from any tab in the Display Grid will allow you to save the Display Grid contents to a CSV file.



Select the Export/Import option.



Note that there is a check box in the upper left allowing you to select all columns to be exported. You can also individually select what columns you wish to export.

Once you click the Export CSV button, the file will be downloaded to your computer. The CSV file can be opened and viewed by many applications including Microsoft Excel.

The CSV can be edited and uploaded back into the system if desired. Using the process above, drag the edited CSV file into the "Load CSV data file" box and click the Import to Grid button.

Server Functions Page

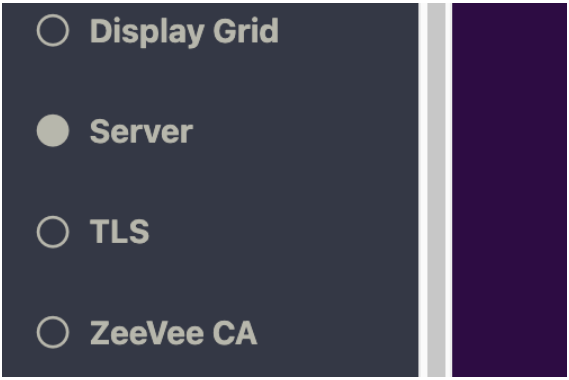
The majority of this page is used to display general information about the Management Platform. In addition, this page provides controls for setting the EDID mode and rebooting or restarting the server. Server firmware can also be updated from this window. For detailed instructions on how to update the ZMP, please refer to appendix 5

Server Information

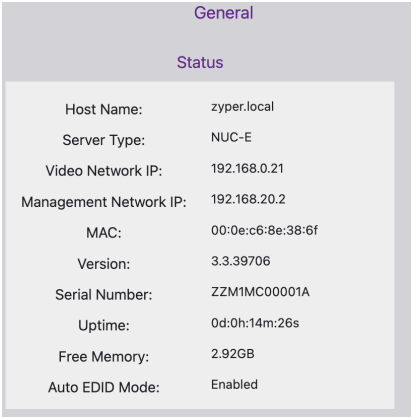
This section provides information about the Management Platform, such as the host name, IP address, MAC address, version, serial number, uptime duration, and free memory.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server option at the left of the page.



3. Information about the Management Platform will be displayed in the Server pane in the General tab.



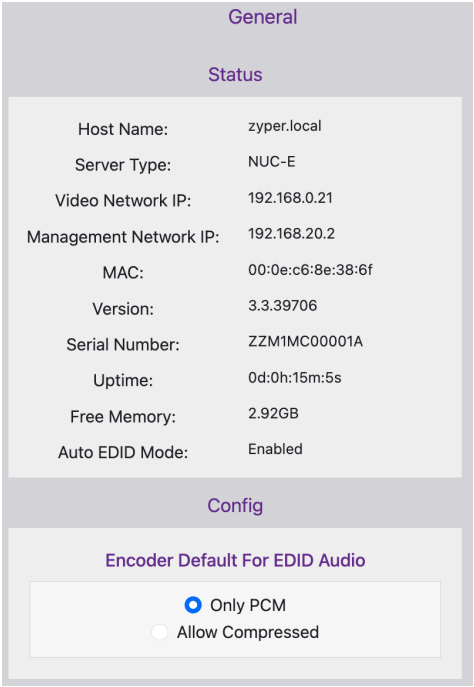
Setting the EDID Audio Mode

EDID Audio setting will specify at the encoder if only raw PCM audio is acceptable to the system or if encoded/compressed audio formats are supported.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server tab at the left of the page. Then select the General tab.

3. EDID information will appear in the Server window under the Status information. You may need to scroll down in the window to see the EDID information.



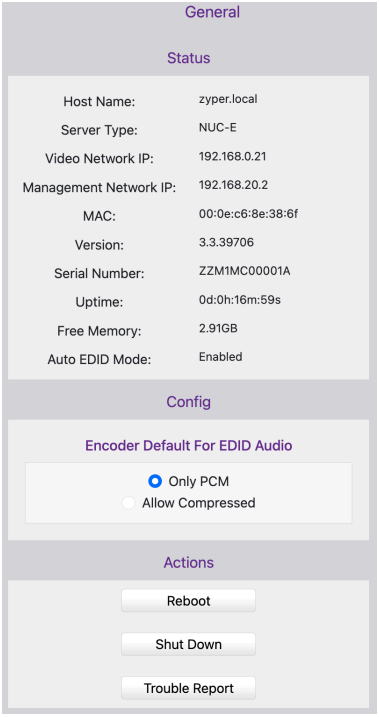
Server Reboot, Shutdown and Trouble Report

The Management Server can be rebooted or shut down from the Server window. Each option affects the Management Server in different ways, as listed below.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server tab at the left of the page. Then select the General tab.

3. Reboot and Shutdown options will appear in the Server window under the Status information. You may need to scroll down in the window to see these options.



4. Click the desired button under Actions.

   ► Reboot
   Linux is rebooted.

   ► Shut Down
   Shuts down the Management Server.

   ► Trouble Report
   Generates a trouble report than can be provide to ZeeVee support.

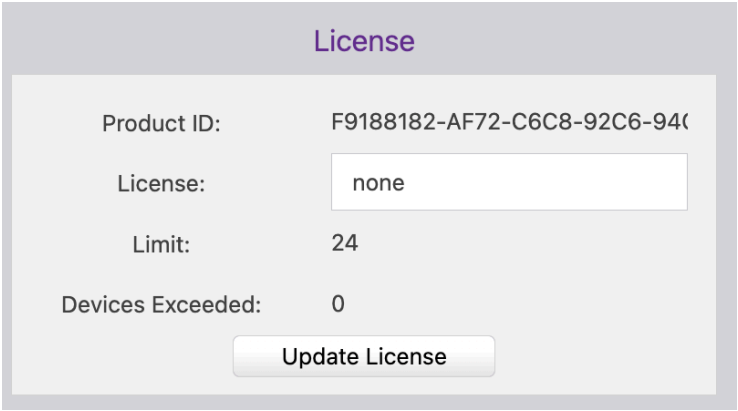
Server License and Update

The License for the Management Server can updated to increase the maximum number of supported endpoints and their Server software version can also be updated.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server tab at the left of the page. Then select the General tab.

3. Scroll down to the License option..



4. Record the Product ID number and provide this to ZeeVee. This ID is used by ZeeVee to generate the new license key when purchased by a customer.

5. Once received, you can enter a new license key as provided by ZeeVee to increase the limit on the number of endpoints. (Please contact ZeeVee support for additional information)

6. Please see the Appendix of this document for information on updating the Server Software. Note: In a redundant environment, Software must be updated on slave first, then switchover, then update on new slave. See online help section 15.2

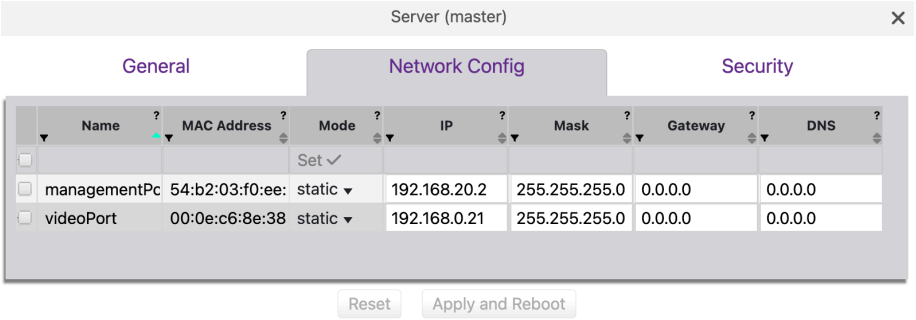
Note: License's are removed starting with release 4.1 of the ZMP API. There is no longer a limit on number of endpoints. Suggested maximum endpoints for NUC hardware is 120. If greater then 120 endpoints it is recommended to use the Rack Mount Server version of the hardware.

Server Network Config

The IP Address of the server Management Port or Video Port can be set in this tab. Note that some older Management Servers only have a single network port. This would be the Video Port. All newer Management servers have both.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server tab at the left of the page. Then select the Network Config tab.



3. You can set either the Management Port or Video Port IP address to DHCP or Static configuration. Always take care when entering a Static address.

4. Once the new information is added, press the "Apply and Reboot" button.

5. The server will now reboot. This may take 1-2 minutes. You will need to login again when prompted.

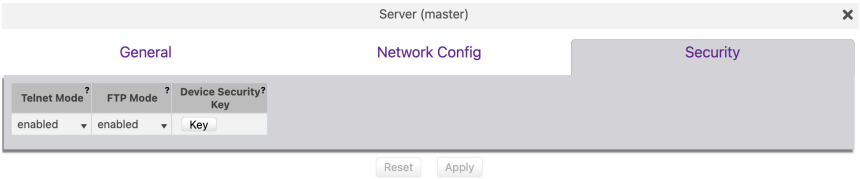
Server Security

The Management Server can enable or disable features for security reasons.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Server tab at the left of the page.

3. Click the right side tab labeled Security



4. From this tab; Telnet and FTP access to the management server can be enabled or disabled. Setting to disable will terminate any active connection and prevent future connections.

5. A server device security key can also be set from this page. The entry must be from 8 to 64 characters in length. This feature is only used with ZyPer4K-XS/XR devices.

TLS

This page and associated tabs shows information regarding security and encryption topics including Certificate Signing Request (CSR), Transport Layer Security (TLS), Certificate Authority (CA), Fullly Qualified Domain Name (FQDN) and Privacy Enhanced Mail (PEM).

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the TLS tab at the left of the page.

Terminology:

CSR (Certificate Signing Request) is the message that's sent to the CA in order to get a digital certificate created. A CSR is often generated on the same server on which the certificate is to be installed.

CA (Certificate Authority) is a trusted entity that issues Secure Sockets Layer (SSL) certificates. These digital certificates are data files used to cryptographically link an entity with a public key. Web browsers use them to authenticate content sent from web servers, ensuring trust in content delivered online.

PEM (Privacy Enhanced Mail) files are concatenated certificate containers frequently used in certificate installations when multiple certificates that form a complete chain are being imported as a single file.

TLS (Transport Layer Security) certificate is a digital object that allows systems to verify the identity & subsequently establish an encrypted network connection to another system using the Transport Layer Security (TLS) protocol. Certificates are used within a cryptographic system known as a public key infrastructure (PKI). PKI provides a way for one party to establish the identity of another party using certificates if they both trust a third-party - known as a certificate authority. TLS certificates thus act as digital identity cards to secure network communications, establish the identity of websites over the Internet as well as resources on private networks.

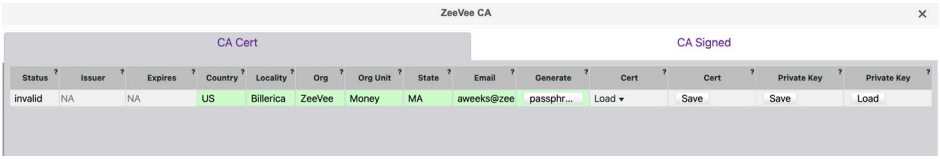
FQDN (Fully Qualified Domain Name) is a complete and unambiguous domain name that specifies an exact location for an object in a Domain Name System (DNS) hierarchy. It specifies all domain levels, including the top-level domain and the root zone.

Please see Section 6 of this manual for more details on Transport Layer Security.

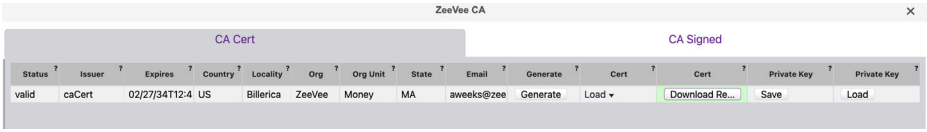
The basic steps are the same regardless of using the web based GUI or the API directly.

1. Setup internal Certificate authority by creating CA Private Key
2. Create CSR and Private Key
3. Use CSR to create Signed Certificate
4. Load Signed Certificate
5. Enable Web Server TLS mode

- Click on the ZeeVee CA tab at the left of the page
- Setup internal Certificate authority by creating CA Private Key (Click Generate; Enter passphrase and items shown in green columns below; then hit Apply)



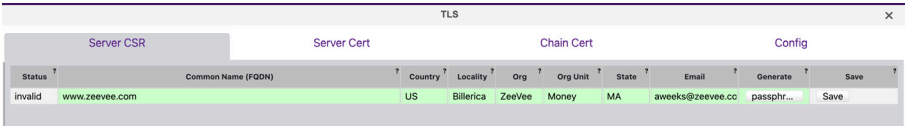
- Save the CA Certificate. Click Save button then Apply. Download will be a ZIP file. caCert.zip Default filename in the zip is caCert.crt



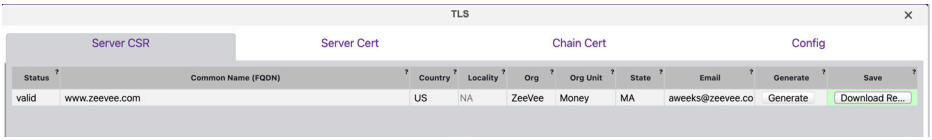
- Save the CA Private Key. Click the Save button then Apply. Download will be a ZIP file caKey.zip Default filename is priv.key

Note: You can also upload the CA Certificate and CA Private key if obtained from an outside source. Then the signed CSR that the ZeeVee internal CA produces is the same as what a 3rd party would produce. Either way, the signed cert is then loaded into the server. (See next page)

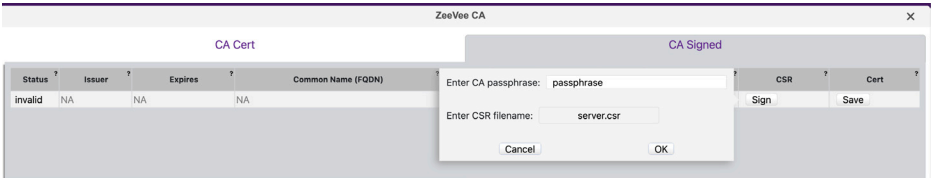
- Use CSR to create Signed Certificate (Click Generate; Enter passphrase and items shown in green columns below; then hit Apply)



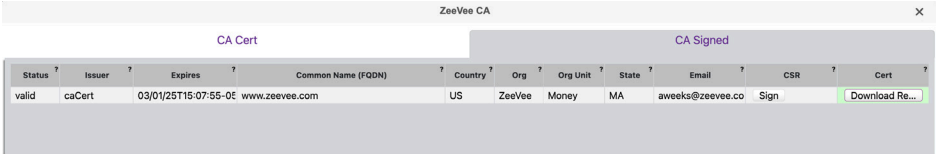
- Save the Server.csr file. This will be in a ZIP file named csr.zip



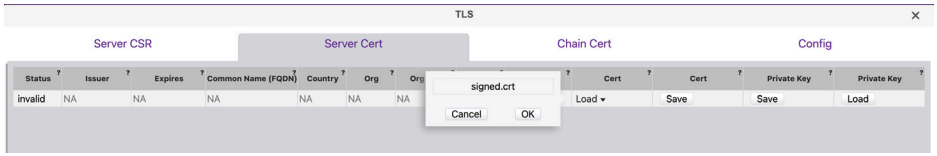
- Sign the Certificate (This is from the ZeeVee CA Tab)



- Save the Signed Certificat. This will be in a ZIP file named signedCert.zip (Default name signed.crt) (From the ZeeVee CA Tab)

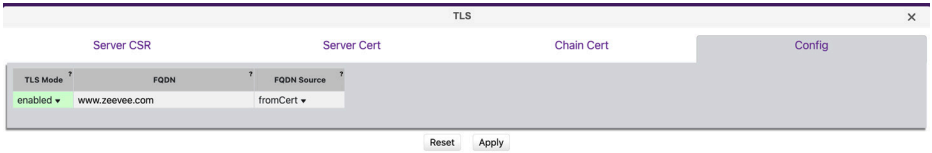


- Load the Signed Certificate

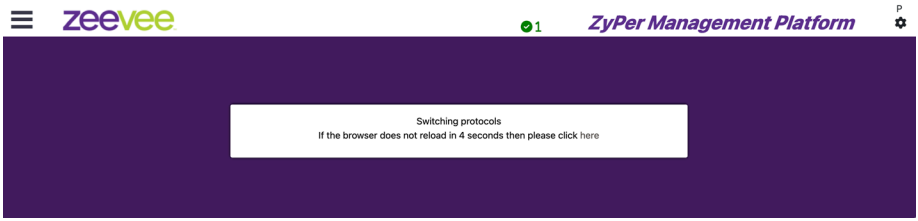


Note: 3rd party certs will always have an additional file: intermediate certs. Those are certs between the signed CSR cert and the CA root cert. That file has to be loaded as well. That file is loaded in the Chain Cert tab)

- Enable Web Server TLS mode



Note: The moment you enable TLS mode you will likely be disconnected from the Server until you take appropriate steps to enable your Browser to once again be able to communicate with the server. Until browser is enabled to communicate properly, you will be directed to the ZeeVee.com website.



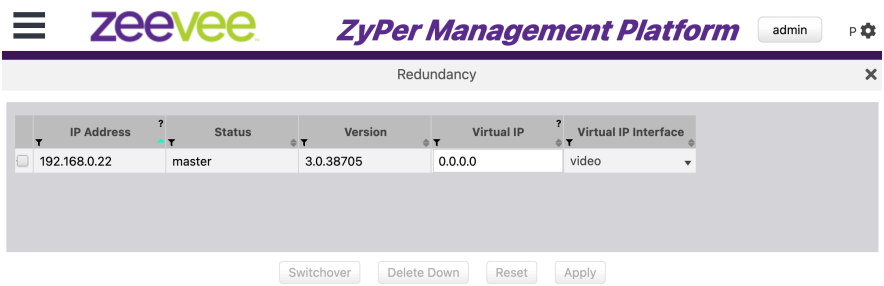
Below is output from Show tls summary command in the API after performing the steps above.

```
Zyper$ show tls summary
server(192.168.0.21);
  server.tls.server; tlsMode=enabled, fqdnMode=fromCert, fqdn=www.zeevee.com
  server.tls.csr; status=valid, C=US, CN=www.zeevee.com, O=ZeeVee, OU=Money, ST=MA,
emailAddress=aweeks@zeevee.com
  server.tls.serverCert; status=valid, C=US, CN=www.zeevee.com, O=ZeeVee, OU=Money,
ST=MA, emailAddress=aweeks@zeevee.com
  server.tls.serverCert; issuer=caCert
  server.tls.serverCert; fingerprint=3025EEEF296FAFDC430FB895A6C741839B3E1C71
  server.tls.serverCert; expires=03/01/25T15:07:55-0500
  server.tls.caChainCert; status=invalid
  ca.tls.caCert; status=valid, C=US, CN=caCert, L=Billerica, O=ZeeVee, OU=Money,
ST=MA, emailAddress=aweeks@zeevee.com
  ca.tls.caCert; issuer=caCert
  ca.tls.caCert; fingerprint=C91A0B6926F892CE8C67E44CA83F28688CA3C365
  ca.tls.caCert; expires=02/27/34T15:04:53-0500
  ca.tls.signed; status=valid, C=US, CN=www.zeevee.com, O=ZeeVee, OU=Money, ST=MA,
emailAddress=aweeks@zeevee.com
  ca.tls.signed; issuer=caCert
  ca.tls.signed; fingerprint=3025EEEF296FAFDC430FB895A6C741839B3E1C71
  ca.tls.signed; expires=03/01/25T15:07:55-0500
  device.tls.csr; status=invalid
  device.tls.deviceCert; status=invalid
  device.tls.deviceCaChainCert; status=invalid
  radius.tls.csr; status=valid, C=US, CN=www.zeevee.com, O=ZeeVee, OU=Money, ST=MA,
emailAddress=aweeks@zeevee.com
Success
```

Redundancy

The Redundancy page provides lets the user manage multiple ZMP units in redundancy mode. This page shows the current configuration and allows changes such as swapping the master ZMP server.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.
2. Click the Redundancy tab at the left of the page.
3. The Management Platform Redundancy window will be displayed.



You can also change the Virtual IP address from this page and select what IP interface port that virtual IP address is viewable from.

Note, older ZMP devices only have a single network interface. (Video Interface)
Newer ZMP devices have two network interfaces. (Video and Management)

Accounts

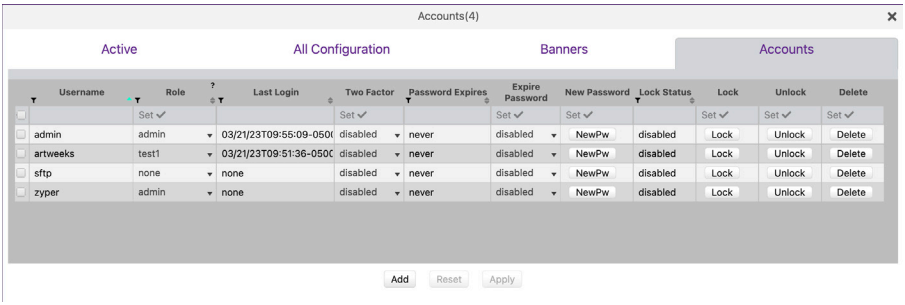
The Accounts page provides the ability to create additional users beyond the Administrator and in conjunction with the Roles page assign different access/abilities to each user.

Other features include the ability to see all Active accounts, set Password requirements and custom Login Banners.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

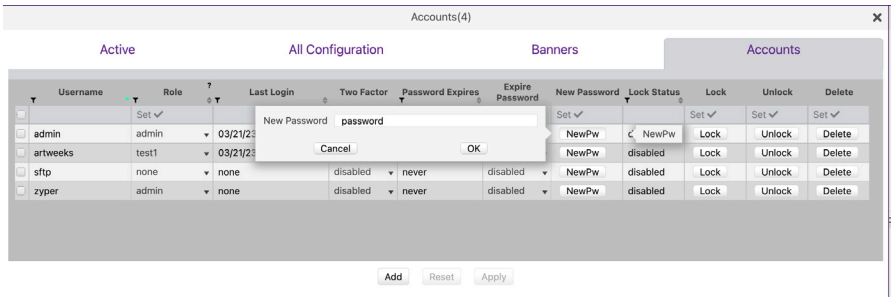
2. Click the Accounts tab at the left of the page.

3. The Management Platform Accounts window will be displayed.



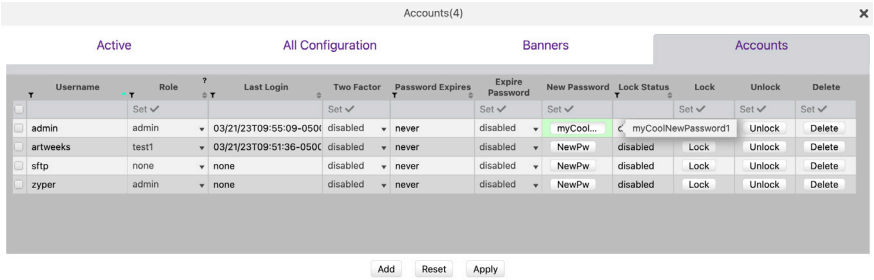
The first User shown in the Accounts window is Admin. The Admin account cannot be deleted and the name cannot be changed. You can however change the password from the default of "admin" to a password of your choosing.

Click on the NewPw button of the New Password column within the Admin user. The following screen will appear and allow you to change the password.



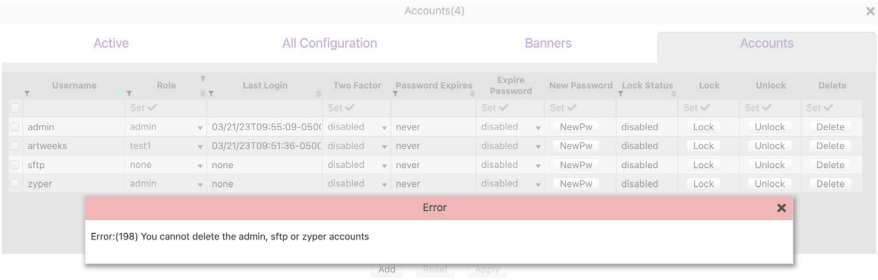
Enter the new password and press the OK button.

The field will now highlight in green and show you the new password. The password will not be implented unit the Apply button is pressed.



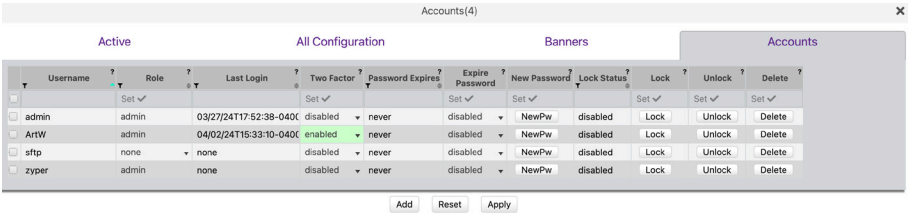
It is recommended to record this new password in a secure location.

Please note that this page allows you to also Lock, Unlock or Delete accounts. The admin, sftp and zyper accounts cannot be deleted. Attempting to do so will result in the following message



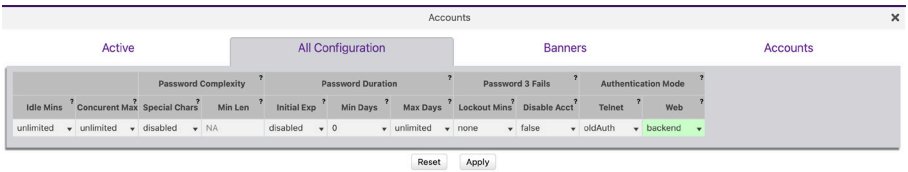
Two Factor Authentication

Two-Factor Authentication is an optional feature that can be enabled for any user account. To enable Two-Factor Authentication, select the desired account in the Accounts tab and set Two Factor to enabled.



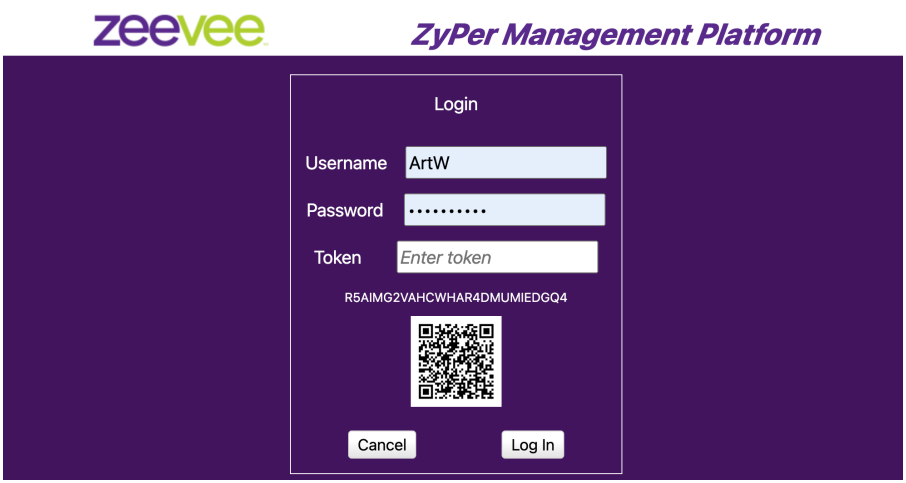
Username	Role	Last Login	Two Factor	Password Expires	Expire Password	New Password	Lock Status	Lock	Unlock	Delete
	Set ✓		Set ✓		Set ✓	Set ✓		Set ✓	Set ✓	Set ✓
admin	admin	03/27/24T17:52:38-0400	disabled	never	disabled	NewPw	disabled	Lock	Unlock	Delete
ArtW	admin	04/02/24T15:33:10-0400	enabled	never	disabled	NewPw	disabled	Lock	Unlock	Delete
sftp	none	none	disabled	never	disabled	NewPw	disabled	Lock	Unlock	Delete
zyper	admin	none	disabled	never	disabled	NewPw	disabled	Lock	Unlock	Delete

YouNote that for Two Factor Authentiation to work; Authentication Mode for the Web must be set to “backend”.



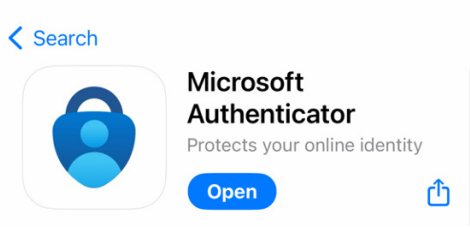
		Password Complexity		Password Duration			Password 3 Fails		Authentication Mode	
Idle Mins	Concurent Max	Special Chars	Min Len	Initial Exp	Min Days	Max Days	Lockout Mins	Disable Acct	Telnet	Web
unlimited	unlimited	disabled	NA	disabled	0	unlimited	none	false	oldAuth	backend

After setting the account for Two Factor Enabled and Authentication Mode to backend, the next attempted login for that user will bring up the speical two-factor authentication screen with the secrect authentication code that can be used with any App Token Generator such as Microsoft Authenticator.

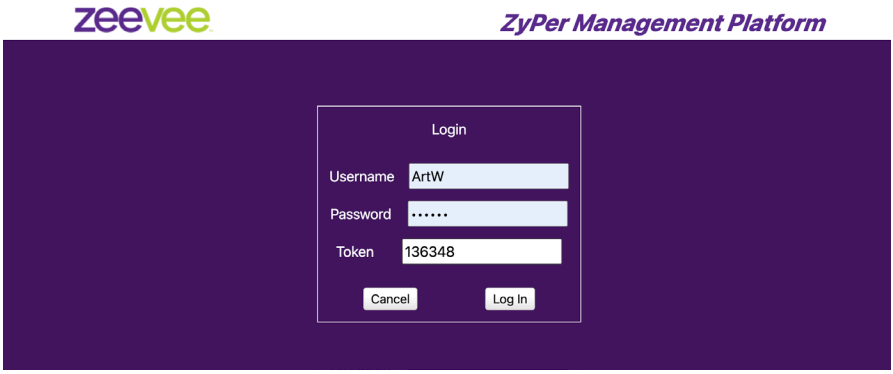


Scan the QR code from Microsoft Authenticator or another application to get token. Note that future login attempts will not show the QR code and will thus be secure.

Microsoft Authencator

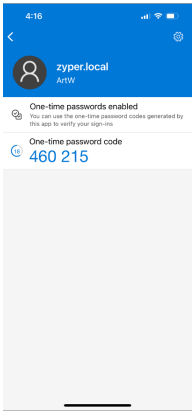


Login attempts after Two-Factor Authentication is configured.
Enter the Token code from Authenticator application.



Important Note: In order for Two-Factor Authentication to work properly; the ZMP itself must have the correct time. (Down to the second).

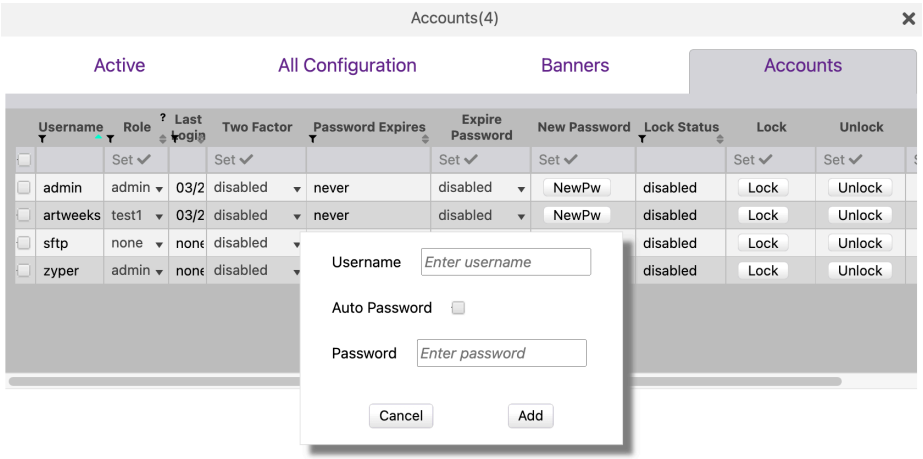
See the set server timezone and set server timezonedate commands in the API section of this manual.



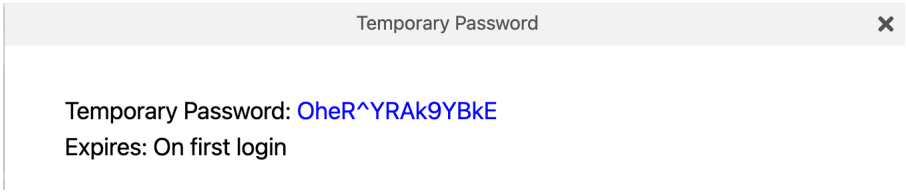
Adding a New User

To create a new user click on the "Add" button in the Accounts window. The screen shown below will appear. Enter the Username and Password for the new user.

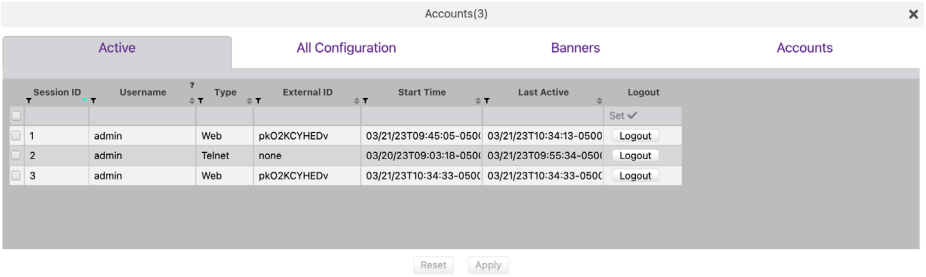
Save the user by clicking the Add button in the lower right corner.



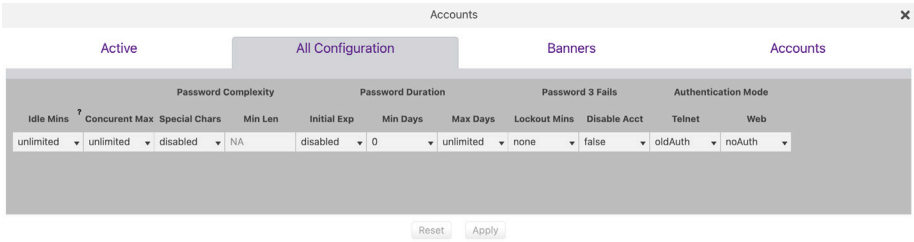
You can have the sysetm create a temporary password automatically by selecting Auto Password. Example below.



Clicking on the Active tab in Accounts will show all accounts currently active (logged in) to the system and allow those with Admin privilages to force a logout from those accounts.



The All Configuration tab provides options for Password Complexity, Password Duration, Password Failures and Authentication mode.



Telnet Authentication options are "oldAuth" and "fullAuth"

oldAuth will not require a telnet access to enter a username and password.
fullAuth does require a telnet access to enter a username and password.

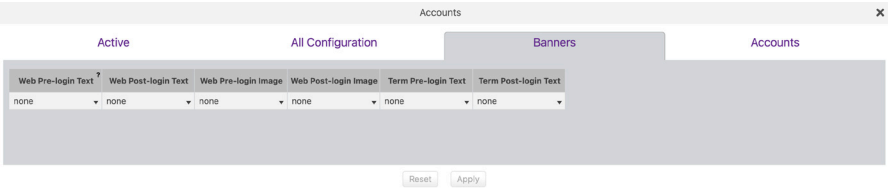
fullAuth Example:

```
telnet 192.168.0.22
Trying 192.168.0.22...
Connected to 192.168.0.22.
Escape character is '^]'.
username: admin
password: *****
Warning:(65) There are 2 other sessions active on this account
Zyper$
```

Note: The ZMP gui does not do browser authentication. However, some users my have an application that does.

To enable backend authentication, the parameter is in the Accounts panel, AllAccounts tab and the AuthenticationMode/Web column.

Clicking on the Banners tab in Accounts will show information about Banners that can be assigned for login or logout of the ZMP.



There are 6 categories as described below:

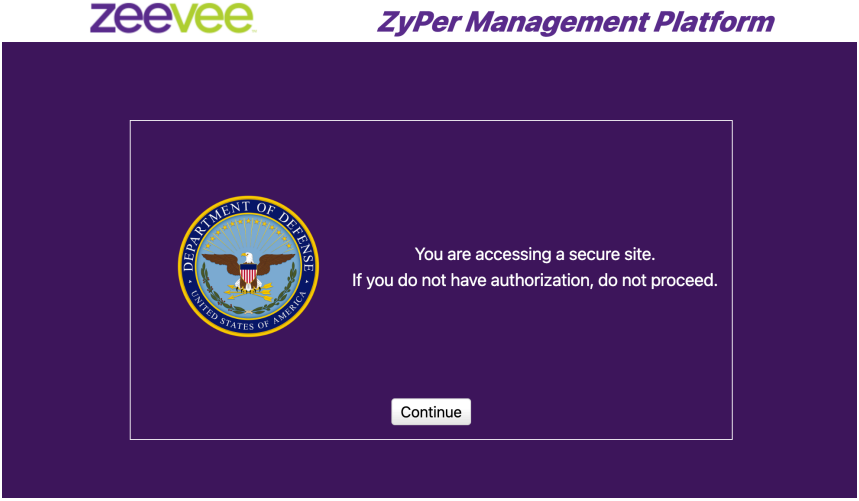
Web Pre-login Text: This is a text message that will appear prior to being allowed to login to the web GUI. (Appears before the login/password screen)

Web Post-login Text: This is a text message that will appear after logging into the web GUI. (Appears after the login/password screen)

Web Pre-login Image: This is a .PNG format image that is displayed prior to being allowed to login to the web GUI. Will be combined with any Web Pre-login text.

Web Post-login Image: This is a .PNG format image that is displayed after logging into the web GUI. Will be combined with any Web Post-login text.

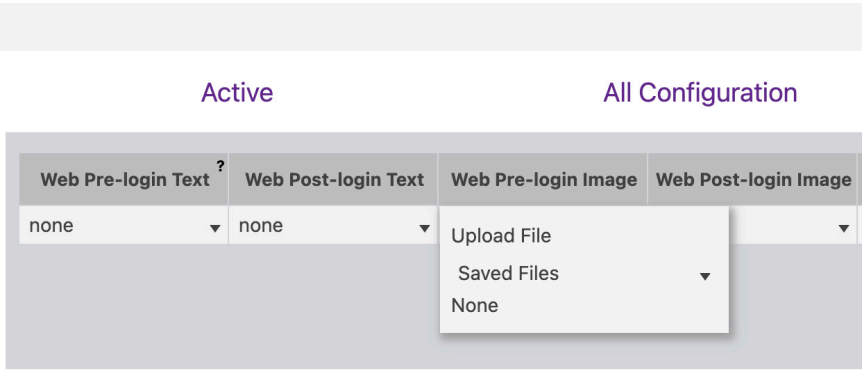
Pre Login Example Below. Combination of image and text:



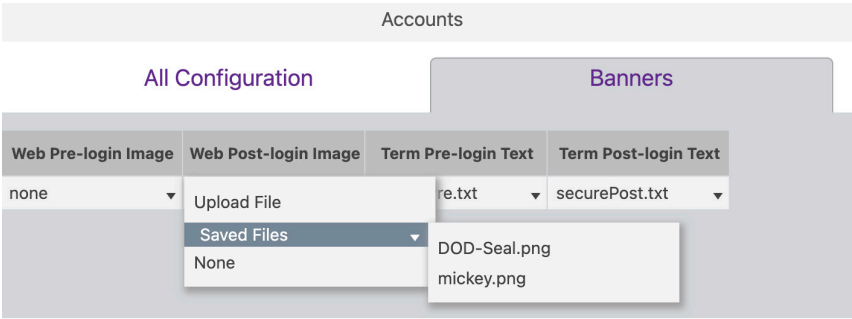
Term Pre-login Text: This is a text message that will appear before you can enter the password during an SSH terminal login.

Term Post-login Text: This is a text message that will appear after you enter the password during an SSH terminal login.

For the text options you can upload a .TXT file or select one that has already been uploaded to the ZMP.



For the image options you can upload a .PNG file or select one that has already been uploaded to the ZMP.



Roles

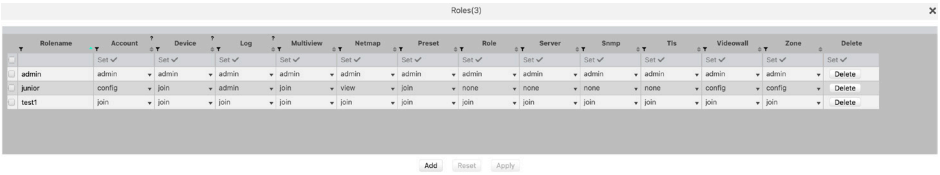
The Roles page provides the ability to create additional users beyond the Administrator and in conjunction with the Roles page assign different access/abilities to each user.

Other features include the ability to see all Active accounts, set Password requirements and custom Login Banners.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Roles tab at the left of the page.

3. The Management Platform Roles window will be displayed.



Adding or Editing a Role

To create a new role click on the "Add" button"+" the Roles window. The screen shown below will appear. In the General tab enter the Role Name. In the example below the name is "Junior_Admin". Note: Spaces are not allowed in Role names.

The next thing you will need to do is assign Permissions to this new role. To do this click on any of the columns.

There are 10 categories that can be assigned different levels of access within the Permissions tab. Account, Device Log, Multiview, Netmap, Preset, Role, Server, SNMP, TLS, Videowall and Zone. Most categories have 5 levels of access:

Admin: User has full functional control over configurations, including ability to delete sources and displays.

Config: User is allowed to alter or create new configurations within the designated tab. For example the user can create a new Wall, Multiview or Zone.

Join: User is allowed to join encoder to decoders or walls. Join multiviews to decoders. Not permitted to create, edit or delete items such as Walls, Mulitviews or Zones.

View: User is allowed to access the designated tab within ZMP but not do anything.

None: No access to this feature. The tabs are not visialbe to the user.

Click on the appropriate levels of access to set permission.

Be sure to Apply any changes when done. Items that will be updated are highlighted in green as shown below.

Help Page

The Help page provides a help reference for each page within the ZMP.

1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

2. Click the Help tab at the left of the page.

3. The Management Platform Help window will be displayed.

Contents

1 What's New In The GUI - Release 4.0.4
2 Sources
3 Displays
4 Join Configurations
    4.1 What are Join Configurations?
    4.2 Pre-defined Join Configurations
    4.3 How are they used?
    4.4 Creating a new Join Configuration
    4.5 Modify a Join Configuration
    4.6 Delete Join Configuration
    4.7 Fast Switched vs. Genlocked
    4.8 Resetting default configurations
5 Preview Streams
6 Video Walls
7 Multiview
8 Zones
9 Presets
10 Source Grid
11 Display Grid
12 Server
13 LDAP
14 TLS
15 802.1x
16 Redundancy (not present on all systems)
17 Accounts
18 Roles
19 Logs
20 How to
22 ZeeVee CA
23 V.A.M. Overview
24 SNMP Users
25 ZMP SNMP Redundancy
26 SNMP Switch Configuration
    26.1 Netgear 4300
    26.2 Cisco Small Business
27 KVM Config

4. Click the desired section. As the mouse pointer moves over each section, the text will become underlined.

3 Advanced Operation

Accessing the API

Using Telnet

Telnet is a popular protocol that can be used on both Windows® and Mac OS® operating systems to connect to the programming shell. On a Windows operating system, a Telnet client, such as "PuTTY", must be installed. From a Unix or Mac OS command line, use the `telnet` command followed by the IP address of the Management Platform:

```
telnet 192.168.0.22
```

> Instead of specifying the IP address of the Management Platform, the following identifier can also be used: `zyper.local`
>
> *Example:* `telnet zyper.local`

Telnet will use port 23 by default and once connected, the API prompt will be displayed:

```
Zyper$
```

Getting Help

To make it easier to find commands, help now supports groups.

- `help` – lists all options
- `help byConcept`-- lists commands within a concept

  Note: The same command may appear in more than one concept.
- `help all bySubsystem` – lists all commands in subsystem
- `help all alphabetical` – list all commands in alphabetical order

Help is available in two forms. Typing `help` or `?` at the prompt will list all available commands:

```
Zyper$ help all alphabetical

 Help All Commands Alphabetical
    add device ipAddress <ip>
    add snmp trapServer v2cTrap ipAddress <address:ip> community
    <string>
    ...
    ...
    update device <deviceNamePart>|all|encoders|decoders <filename>
    update server <filename>
Success
Zyper$
```

```
Zyper$ help

 Help commands:
    help
    help all alphabetical
    help all byConcept
    help all bySubsystem
    help all byAccessLevel
    help concept <helpConcepts>
    help subsystem <helpSubsystems>
    help accessLevel <helpAccessLevels>
    help search string <keyWord:string>
    <command> help
    <command> ?
    ?

** NOTE: Use <tab> to complete a command **

Success
```

In addition, a partial list of commands can be listed by specifying the first word of each command. The first part of the command must be specified before the `help` command. For example, the following will only list command with the join prefix.

```
Zyper$ join help
join help
join <encoderMac|encoderName>|none <decoderMac|decoderName|zoneNa
me[.zoneName]> analogAudio|danteAudio
join <encoderMac|encoderName>|videoSource|none <decoderMac|decoderN
ame|zoneName[.zoneName]> hdmiAudio
join <multiviewName>|none <decoderMac|decoderName> multiview
join <encoderMac|encoderName>|none <decoderMac|decoderName|zoneNa
me[.zoneName]> video|fastSwitched|genlocked|genlockedScaled
join <encoderMac|encoderName>|none <wallName> videoWall
join <encoderMac|encoderName>|none <decoderMac|decoderName>|none
usb
join <encoderMac|encoderName>|none <decoderMac|decoderName> window
viewportSource <x:int> <y:int> <sizeX:int> <sizeY:int> viewportDest
<x:int> <y:int> <sizeX:int> <sizeY:int>
join none all video|analogAudio|hdmiAudio
Zyper$
```

In addition, help can be searched by keyword. help search string <string>

```
Zyper$ help search string layer
set multiview <multiviewName> windowNumber <int> encoderName
<encoderName>|none percentPositionX <float> percentPositionY <float>
percentSizeX <float> percentSizeY <float> layer <int>
set multiview <multiviewName> windowNumber <int> encoderName
<encoderName>|none pixelPositionX <int> pixelPositionY <int>
pixelSizeX <int> pixelSizeY <int> layer <int>
set multiview <multiviewName> windowNumber <int> layer <int>
Success

Zyper$ help search string wall
add zoneDisplay <zoneName[.zoneName]> <decoderMac|decoderName|wall
Name>
create videoWall <newWallName>
delete videoWall <wallName>
delete zoneDisplay <zoneName[.zoneName]> <decoderMac|decoderName|wa
llName>
join <encoderMac|encoderName>|none <wallName> videoWall
set role rolename <rolename> subsystem all|netmap|device|preset
|zone|account|role|videowall|log|multiview|server|snmpagent|tls
maxAccess none|view|join|config|admin
set videoWall <wallName> newName <newWallName>
set videoWall <wallName> decoder <deviceMac|deviceName>|none row
<int> column <int>
set videoWall <wallName> size rows <int> columns <int> topBezel
<int> bottomBezel <int> leftBezel <int> rightBezel <int>
show videoWalls
Success

Zyper$ help search string create
create account <newUsername> password <string>|*
create account <newUsername> tempInitialPassword
create multiview <newMultiviewName>
create presetNew <newPresetName> commands existingConnections|empty
create presetSchedule <presetName> schedule <newPresetScheduleName>
create role <newRolename> allSubsystems maxAccess
none|view|join|config|admin
create videoWall <newWallName>
create zone <[zoneName.]newZoneName>
Success
```

Setting the Time Zone

The Management Platform can use the Network Time Protocol (NTP) to set the date and time. However, the time zone will need to be specified. Alternately the date and time can be set manually.

1. Telnet to the Management Platform.

```
telnet 192.168.0.22
```

2. After the connection has been established, use the `set server timezone` command to set the time zone.

> The time zone must be specified in POSIX format and is case-sensitive.
> Refer to the following link for more information:
>
> http://wikipedia.org/wiki/List_of_tz_database_time_zones.

```
Zyper$ set server timezone America/New_York
Success
Zyper$
```

3. To set date/time manually use the `set server timezonedate manual` command.

```
Zyper$ set server date manual  month 4 day 1 year 2021 hour 15
minute 1
Success
Zyper$
```

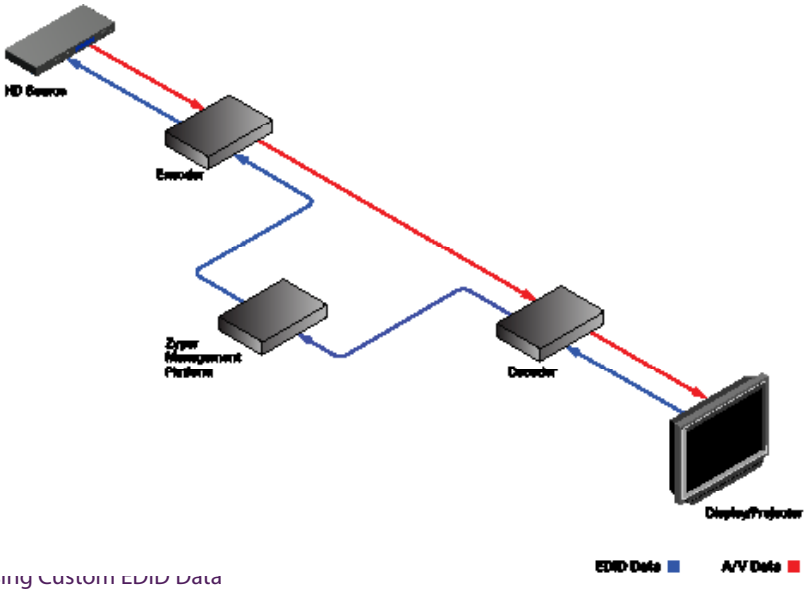
Use the `show server info` command to verify the correct time zone has been set.

```
Zyper$ show server info
server(192.168.0.22);
  server.gen; hostname=zyper.local, serverType=NUC-D, version=3.2.39285,
 previousVersion=3.2.39280, master=true
  server.gen; uptime=2d:1h:23m:58s, freeMem= 6.43GB, sdvoeVersion=3.5.0.0,
 bootCount=11, serialNumber=ZZM1K400011D
  server.gen; macAddress=94:c6:91:a0:47:fc, managementMacAddress=
  server.ipActive; ipServerAddr=192.168.0.22, ipManagementAddr=192.168.0.22,
 gatewayAddr=192.168.0.1, dnsAddr=0.0.0.0
  server.ipActive; managementGatewayAddr=0.0.0.0, managementDnsAddr=0.0.0.0
  server.time; time="Thu Oct  5 11:25:02 2023", timezone=America/New_York
  server.license; productID=F9188182-AF72-C6C8-92C6-94C691A047FC,
 license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=7, allDevices=19,
 allDevicesUp=8, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_1.7.15.0=1, num_1.7.23.0=2,
 num_1.7.24.0=4, num_1.7.4.0=1
lastChangeIdMax(165);
Success
```

EDID Management

Auto EDID Mode

By default, Auto EDID mode is enabled. This means that the Management Platform will compare the encoder EDID with the decoder EDID. If they are different, then the EDID from the decoder (sink) will be used by the encoder (source). Setting the EDID Mode affects all join modes: fast-switched, genlocked, and video-wall. Refer to the `join` command in the API Command Listing (page 101) section for more information.



Using Custom EDID Data

There may be some instances where a custom EDID is desired. One example is when using a single encoder with multiple displays, such as a video wall. In such a case, follow the steps below to save and load a custom EDID to the Management Platform.

1. Telnet to the Management Platform.

   ```
   telnet 192.168.0.22
   ```

2. Disable Auto EDID mode by entering the following command:

   ```
   zyper$ set server auto-edid-mode disabled
   ```

3. Use the `save deviceEdid` command to save the EDID of the sink device (attached to the decoder) to the Management Platform, using the following convention:

   ```
   save device-edid [id] [filename]
   ```

   Make sure to replace `[id]` with the identifier of the sink device containing the EDID you wish to capture. You can specify either a MAC address or a name identifier. Follow the identifier with the name of the EDID file. For example:

   ```
   zyper$ save device-edid SonyXBR4 myEDID
   ```

4. After executing this command, two files will be created under the following directory:

   ```
   /srv/ftp/files/myEDID
   /srv/ftp/files/myEDID.txt
   ```

   `myEDID` is a binary EDID data file in standard format. `myEDID.txt` contains the decoded EDID in standard ASCII text.

   These files must remain in this directory when disabling Auto EDID mode.

5. To force a ZyPer encoder to use the saved EDID you need to have the MP load the binary EDID file onto the desired encoder.

   ```
   zyper$ load encoder-edid [id] saved [filename]
   ```

   Make sure to replace `[id]` with the identifier of the source device you want to load the EDID onto. You can specify either a MAC address or a name identifier. Follow the identifier with the name of the EDID file. For example:

   ```
   zyper$ load encoder-edid BlueRay1 saved myEDID
   ```

6. To return to Auto EDID mode, for any reason, enter the following command at the prompt:

```
zyper$ load encoder-edid BlueRay1 auto
```

or

```
zyper$ set server auto-edid-mode enabled
```

Using AJAX/JSON

The AJAX/JSON programming interface allows developers to control the Management Platform within browser-based applications. All calls to the server are asynchronous post/receive operations using Javascript and do not require any specific HTML or CSS code. We will present two examples in this section: Login authentication and command request/response.

Login Authentication

There are two methods to authenticate with the server. The first and recommended method is to pass the username and password to `rcLogin.php`. The second method is to pass the username and password in every AJAX request.

Once the server accepts the username and password, it will generate a secure cookie called "userToken". This cookie will expire one hour after the last AJAX command is received by the server. After the cookie expires, all other AJAX requests will result in a failed authentication until `rcLogin.php` is called again. The following code excerpt is from the `zyperLogin()` function within `zyper.html`:

```
...
...
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange = function(){
   if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
      procLoginResp(xmlhttp.responseText);
      }
   }
postdata = "";
postdata += encodeURIComponent("serverSocketName") + '=' +
            encodeURIComponent(socketName) + '&' +
            encodeURIComponent("username") + '=' +
            encodeURIComponent(username) + '&' +
            encodeURIComponent("password") + '=' +
            encodeURIComponent(password) + '&';
xmlhttp.open("POST", url, true);
xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
xmlhttp.send(postdata);
}
```

The response is a string value. The variable `resp` can be "Success", "Failed", or "Server not running".

```
function procLoginResp(jsonData) {
       var resp = JSON.parse(jsonData);
...
...
```

Command Request / Response

After login, any further commands are sent to the rcCmd.php

The following code excerpt sends an AJAX request to list all ZyPer encoders and decoders:

```
function zt(){
  xmlhttp = new XMLHttpRequest();
  xmlhttp.open("POST", url, true);
  xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
  xmlhttp.onreadystatechange = function(){
     if (xmlhttp.readyState == 4 && xmlhttp.status == 200){
        procResp(xmlhttp.responseText);
     }
  }
xmlhttp.send(encodeURIComponent("commands:show device-status
all"));
}
```

In this example, the `encodeURIComponent` function has two parts: The request type, which is `commands` and the command `show device-status all`. Refer to the `show device status` command for more information. Currently, `commands` is the only request type that is supported and only a single command can be suppplied for each request.

Here, we handle the AJAX response:

```
function procRespTest(jsonData){
   var jsData = JSON.parse(jsonData);
   # jsData.status may have the values:
   # "Success"
   # "Request failed authentication"
   # "Server not running"
   # "no commands provided"
   #
   if (jsData.status == "Success"){
      var element = document.getElementById("responseError");
      element.innerHTML = jsData.responses[0].error;
      element = document.getElementById("responseWarning");
      element.innerHTML = jsData.responses[0].warning;
      element = document.getElementById("numObjectsInResponse");
      element.innerHTML = jsData.responses[0].text.length;
   }
   else{
      // Failed authentication
   }
}
```

The JSON data is decoded using the `JSON.parse()` method. In this example, information about the response data is displayed on the web page (HTML code not shown).

The JavaScript object that is returned is:

```
var jsObj = {
   status: true | false;
   responses: [ {error: "errorText",
                 warning: "warningText",
                 text: [ { param1: "val1", parmN: "paramN" } ]
                 }
              ]
   };
```

The return value is an object that contains two members: `status` and `responses`. If the `status` member is not equal to "Success", then the `responses` member is not valid. If the request fails authentication, then the `status` value will be "Request failed authentication". Note that there may be other web-server level failures that can be returned in the `status` string.

The second member in the returned object, `responses`, which is an array of objects. Each of these objects contains three members: `error`, `warning`, and `text`. The `error` and `warning` members are strings. The `text` member is an array of objects with the desired parameters and values. If the `error` string is non-null, then the `warning` and `text` members will be null. If `text` is non-null, then the `warning` string may still be valid.

Currently, the `responses` member is always an array size of 1.

Note: ZeeVee provides a file zyper.html provides an example of all web-access commands.

From Browser the file can be accessed using IP Address of the ZMP.

http://<ipAddr>/zyper.html

Fast-Switched vs. Genlocked Mode

The ZyPer4K provides two uniquely different modes for joining video/audio between a source (encoder) and display (decoder). The chart below details the differences between these two modes.

Feature	Fast-Switched	***Genlocked***
Latency	1-frame of latency. (16-33ms depending on frame rate of source video)	0 frames of latency. Less than 100µs
Transition Appearance	Instantaneous if switching between sources at same resolution and frame rate	Visible blanking of display when switching between sources
Scaling	Automatic scaling up or down to preferred resolution of the display (As determined by display EDID)	Source is not scaled. What comes in at source is presented to display exactly as input. (Note: Special Genlock-scaled mode is available)
HDR	HDR input is automatically reduced to 8-bits at output	HDR input is maintained exactly as input at the output
Color Space	Output from decoder is always RGB	Output from decoder matches the input at encoder
Encoded Audio	AC3 or other encoded audio formats are passed from encoder to decoder	AC3 or other encoded audio formats are passed from encoder to decoder
Video Wall	Video walls are technically not supported in Fast-Switched mode. (Join command for walls defaults to Genlock-scaled)	Video walls are technically always in Genlock-scaled mode
Multiview	Multiview is supported in Fast-Switch mode	Multiview is not supported in Genlocked mode
Video Disconnect	Disconnecting video (Join None) will maintain a black screen output. (Video is not technically disconnected)	Disconnecting video (Join None) will disconnect the video stream entirely. No video output from decoder
USB, IR, RS232	None of these items are associated with Fast-Switched or Genlocked mode	

Notes: Video Disconnect refers to the existing join between encoder and decoder. If existing join is Fast-Switched then a "Join None" command simply puts out a fully black video output from the decoder. If the existing join is Genlocked, then a "Join None" command disconnects the video stream and nothing is output from the decoder. This can be verified by looking at the VID light on the ZyPer4K decoder.

API Command Listing

Command	Description
`add device`	Manually adds a device to Management Platform
`add ldap`	Add LDAP Group
`add snmp`	Add new snmp user or trap server
`add snmp netNode`	Used to add Netnode. (VAM feature)
`add zoneDisplay`	Adds a display or Video wall to an existing zone.
`authenticate username`	Used by browser to authenticate users
`channel`	Cycles up/down through encoders. Used to change channels.
`clone multiview`	Used to clone an existing multiview.
`convertPid`	Used to manually update Product Identification code on 12G SDI model encoders. (ZyPer4K family only)
`create account`	create a new user account with password
`create kvm`	create a new kvm
`create multiview`	Creates a new multiview display (ZyPer4K family only)
`create presetNew`	Creates a new preset
`create presetSchedule`	Creates schedule for existing preset
`create role`	Create a new role with specified access level
`create videoWall`	Creates an empty 2x2 video wall.
`create zone`	Creates a new empty zone.
`dataConnect`	Used Creates a TCP port connection between devices for IR or RS232
`delete account`	delete a user account
`delete allConfiguration`	Deletes all encoder/decoder and server information from the Management Platform
`delete device`	Used Deletes the specified encoder or decoder from the Management Platform database.
`delete kvm`	Deletes the specified kvm from the Management Platform database
`delete ldap`	Delete LDAP Group
`delete multiview`	Deletes the specified multiview from the Management Platform database. (ZyPer4K family only)
`delete multiviewWindow`	Deletes a specific window from an existing multiview (ZyPer4K family only)
`delete preset`	Used to delete a preset, preset runlog or preset schedule
`delete role`	Delete an existing role
`delete snmp`	Delete SNMP user or trap server

Command	Description
delete snmp netNode	Delete SNMP Netnode (VAM feature)
delete videoWall	Deletes the specified video wall from the Management Platform database.
delete zone	Deletes an existing zone
delete zoneDisplay	Removes a display from an existing zone
diagnostics device	Used Runs a set of diagnostics on device
dumpusb	Outputs information about USB devices
events	Causes the event mode to be entered
factoryDefaults device	Sets the specified encoder/decoder to factory-default settings.
flashLeds	Plysically identifies the specified encoder/decoder on the network using LED flashes.
generate tls ca privKeyPass	Used to generate a local Transport Layer Stream Certificate Authority private key.
generate tls server/device csr privKeyPass	Used to generate a local Transport Layer Stream server Certificate Signing Request private key.
help	Brings up various help options
join	Switches audo and/or video from source to display or video wall
join videoSource	Selects audio feed to follow a video join
load account	Sets GUI pre and post login images and warning text
load encoderEdid	Uploads an EDID file to the specified encoder
load idleImage	Uploads an image to use as UHD background when no video streamed to decoder (ZyPerUHD only)
load tls	Used to load TLS certifications and keys
logging	Used to set logging level and add notes to the log
logout	Used to logout current session or force logout of any active session
previewStream	Used to turn on/off the preview stream viable in the Management Platform GUI (ZyPer4K, ZyPerUHD60 and ZyPerUHD only)
redundancy add server ip	
redundancy delete downServers	Removes no longer present servers from list of redundant servers
redundancy delete server ip	
redundancy switchover	Swap Management Platform Master and Slave
rename kvm	Used to change the name of a KVM
rename zone	Used to change the name of a Zone
restart device	Restarts the specified encoder/decoder
restore server database	Restores a saved database

Command	Description
`revert server`	Switch to a previously installeed version of the API
`save deviceEdid`	Saves the EDID from a decoder to a local file
`save server database`	Saves current server database to file
`save system config`	Saves current system configuration to a file
`script`	Executes the specified AJAX/JSON or text script.
`send`	Sends an IR, CEC or RS232 string to the specified device
`set account`	Sets various security features
`set decoder connectionMode`	Changes current connection to decoder to fastSwitched, genlocked or genlockedScaled. (ZyPer4K family only)
`set decoder displayMode`	Sets defaults decoder output to crop, stretch or box
`set decoder displayResolution`	Used to set decoder output size to auto or manual resolution. (Width, Height, FPS)
`set decoder analogAudioOut source`	Sets the source of Analog audio output for specified decoder
`set decoder danteAudioOut`	Sets source for audio to put on Dante network
`set decoder edidPreferMode`	Sets the preferred resolution from the display EDID
`set decoder hdmiAudioOut source`	Sets the source of HDMI audio output for specified decoder
`set decoder hdmi5vControl`	Enables or disables HDMI 5V line (ZyPer4K-XS/XSE and ZyPer4K-XR only)
`set decoder lowLatency`	Enables or disables low latency mode on ZyPerUHD60 (ZyPer4UHD60 only)
`set decoder osdStatusMode`	Enables or disables OSD feature (ZyPerUHD and ZyPerUHD60 only)
`set decoder powerSave`	Enables or disables power-save feature (ZyPerUHD and ZyPerUHD60 only)
`set device general name`	Sets the name for the specified device.
`set device ip dhcp linkLocal`	Sets the specified device to DHCP or Link-Local mode (ZyPer4K family only)
`set device ip static`	Sets the device to static mode (ZyPer4K family and ZyPerUHD only)
`set device dante port`	Sets the device Dante port to Utility or Video (ZyPer4K-XSE family and ZyPerUHD60-2EA/2DA only only)
`set device dante ip dhcp linkLocal`	Sets the specified device dante core to DHCP or Link-Local mode (ZyPerUHD60-2EA/2DA only)
`set device dante ip static`	Sets the device dante core to static mode (ZyPerUDH60-2EA/2DA only)
`set device dante vlan mode`	Configured Dante VLAN mode and ID (KDS-17 family only)

Command	Description
`set device irProcessing`	Configures decoder to process incoming IR commands to issue Channel up/down command (ZyPer4K family white boxes only)
`set device rs232`	Sets the RS232 settings for the specified device
`set device optionCard`	Manually set ZyPer4K extended input encoder functionality
`set device security`	Enables security between server and device (ZyPer4K-XS/XSE and ZyPer4K-XR only)
`set device sendIpMcastRange`	Sets allowable range of multicast addresses for selected devices (ZyPer4K family only)
`set device sourceDisplay location`	Sets the location name for the specified device.
`set device sourceDisplay manufacturer`	Sets the manufacturer name for the specified device.
`set device sourceDisplay model`	Sets the model name for the specified device
`set device sourceDisplay serialNumber`	Sets the serial number name for the specified device
`set device usbFilter`	Allows restrictions to USB use on selected device (ZyPer4K family only)
`set device usbType`	Enable or disable Icron USB (ZyPer4K-XSE only)
`set device utilityPort`	Enables or disables the 1G Ethernet utility port for the specified device (ZyPer4K and UHD60 family only)
`set device videoPort`	Selects active input port for ZyPer4K units with multiple inputs (ZyPer4K family only)
`set encoder analogAudioOut source`	Sets the source of Analog audio output for specified encoder (ZyPer4K and UHD60 family only)
`set encoder edid audio`	Sets allowable input audio formats
`set encoder hdcpMode`	Sets the HDCP compatibility at the encoder side
`set kvm`	Configure kvm positions, USB source and Hot Keys
`set ldap mode`	Enable or disable LDAP support
`set ldap server address`	Set the domain name of LDAP server
`set ldap server`	Controls LDAP settings
`set ldap type`	Sets LDAP to either openLdap or ActiveDirectory
`set multiview`	Assigns source to a position and size within a multiview display (ZyPer4K family only)
`set multiview allowMainStream`	Determines if main unscaled stream can be used in a multiview (ZyPer4K family only)
`set multiview audioSource windowNumber`	Selects the input source to provide Audio for multiview display (ZyPer4K family only)
`set multiview canvasSize`	Specifies Multiview canvas for multiview creation. (ZyPer4K family only)

Command	Description
`set multiview newEncoderName`	Used to specify a new encoder for existing multiview window. Can also set to “none”
`set preset commands`	Specifies commands to be used for a preset
`set preset description`	Sets a description for the preset
`set preset schedule eventColor`	Sets the color to be used for a preset schedule in the GUI calendar
`set preset schedule month`	Sets the schedule month/day/time to run preset
`set responses rs232TermChars`	Specifies the RS232 termination string
`set role`	Sets permisison levels for a specific role.
`set server api lineWrap`	Sets number of characters before API command line interface starts a new line
`set server autoEdidMode`	Sets the EDID mode
`set server contact`	Sets server contact information
`set server dataTunnelMode`	Sets server transfer mode to raw or telnet
`set server date`	Used to set server date manually or via ntp server
`set server discoverMode`	Used to set how server discovers ZyPer endpoints. Broadcast or Multicast
`set server encoderDefault edid audio`	Sets the default encoder audio format for HDMI audio input.
`set server ftp mode`	Enables or Disables FTP access to Management Platform
`set server hostName`	Sets server Hostname
`set server isolationMode`	Used to enable/disable auto discovey of endpoints and servers
`set server ip`	Sets the IP address of the Management Platform
`set server isaac address`	Sets the domain name of the Isaac server
`set server isaac subsystemId`	Sets the subsystem ID of the Isaac server
`set server license`	Sets server license. (Controls max endpoints)
`set server location`	Sets server location
`set server redundancy`	Set a virtual IP address/mask for Master and Slave Management Platforms
`set server security`	Set server device Security Key. (ZyPer4K-XS/XSE and ZyPer4K-XR only)
`set server telnet mode`	Used to enable or disable telnet access
`set server telnet password`	Used to set telnet password
`set server timezone`	Sets the time zone

Command	Description
`set terminal output`	Select either normal or JSON format output from API
`set tls`	Used to enable web server TLS mode
`set snmp netNode`	Used to change netNode settings of Nodes previously added. (VAM feature)
`set snmp netNode ipAddress`	Used to change netNode settings of Nodes previously added. (VAM feature)
`set videoWall`	Modifies an existing wall
`set videoWall Decoder`	Assigns the specified decoder to a position within the video wall
`show account`	Shows information about accounts
`show dataTunnels`	Shows what rs232 or IR data relay ports are opened on the server.
`show device capabilities`	Shows detailed capabilities of specified device or devices
`show device config`	Shows detailed configuration information for specified device or devices
`show device connections`	Shows encoder connections to decoders
`show device names`	Shows device names, MAC address, IP address and current state
`show device status`	Provides detailed status information for specified device or devices
`show device userAdded`	Will show a list of all ZyPer endpoints that have been manually added with the add device command
`show multiviews config`	Lists all created multiviews with source, position and size info (ZyPer4K family only)
`show multiviews status`	Lists all created multiviews with source, datarate and multicast address info (ZyPer4K family only)
`show files`	Show various types of files currently stored on Management Server. (EDID, Firmware, Icons, Idle Images)
`show kvm config`	Shows a listing of configured kvms
`show kvm swapped`	Shows a listing of current kvm status
`show ldap groupIdToRole`	r
`show ldap server`	Shs
`show logs commands`	Shows a listing of last commands send to the Management Server
`show logs authentications`	Shows listing of recent logon/logoff events
`show preset`	Shows information and configuration details for a preset
`show previewStreams`	Lists names of encoders currently generating a preview stream. (ZyPer4K family and ZyPerUHD family)

Command	Description
`show responses`	Displays the lastChangeId for the specified device
`show role`	Shows information about a specific role or all roles
`show server config`	Displays the IP address and EDID mode of the Management Platform
`show server info`	Displays Management Platform information
`show server ip duplicates`	Shows cases were an IP address has been duplicated in the system. (Issue needs to be resolved)
`show server redundancy`	Displays information about Master and Slave Management Platforms
`show server managedDevices`	
`show snmp`	Displays information related to SNMP
`show snmp netNode`	Shows information related to netNodes used with VAM feature
`show tls ca pem`	Show TLS Private Enhanced Mail details
`show tls device summary`	Show TLS Device summary
`show tls summary`	Show Transport Layer Security summary
`show values`	Shows information related to encoder, decoder, server and multiviews
`show videoWalls`	Displays a list of all created video walls
`show zones`	Displays a list of zone and displays contained within
`shutdown server`	Shuts down or reboots the Management Platform
`sign tls`	Used with CSR to create signed certifiate
`sleep`	Sets a time delay, in milliseconds
`start encoder`	Start a specified stream (ZyPer4K family only)
`start kvm`	Start a specified kvm (ZyPer4K XS/XSE/XR only)
`stop encoder`	Stop a specified stream (ZyPer4K family only)
`stop kvm`	Start a specified kvm (ZyPer4K XS/XSE/XR only)
`switch`	Switches IR or RS-232 between devices
`troubleReport`	Generates a trouble report
`update device`	Updates the individual encoder or decoder units
`update server`	Updates the Mangement Platform software. See Updating the Software (page 412) for more information

add device

Used to manually add a device to the ZyPer Management Platform that are located on a different VLAN/Subnet than the ZMP itself. Also used to add devices to ZMP when in Isolation Mode.

A qualified network engineer should be involved in making these configuration updates and the network switch provider may need to be consulted to ensure support of needed features.

Syntax

```
add device ipAddress ip
```

Parameters

i

Type: IP Address

The IP address of the device

Example

```
add device ipAddress 192.168.10.81
Success
```

Detailed Example

The ZyPer4K Endpoints are located on VLAN 10 and the 192.168.10.X subnet. The ZyPer Management Platform is on VLAN 20 and the 192.168.20.X subnet.

The ZMP will automatically discover any ZyPer4K endpoints located on VLAN 20. The ZMP will NOT automatically discover any ZyPer4K endpoints located on VLAN 10. However, given the proper circumstances, the ZyPer4K endpoints on VLAN 10 can be manually added to the ZMP for control.

For this to work, the network MUST be configured to route traffic between VLAN 10 and VLAN 20. How to configure the network to allow routing between VLANs is beyond the scope this document and should be done by a qualified network engineer. A simple test to confirm routing is that a device in VLAN 10 can ping a device in VLAN 20.

The ZyPer4K endpoints need to have a known IP Address. The IP Address should either be assigned by a DHCP server or assigned statically.

ZyPer4K endpoints need to be added one at a time.

You can get a listing of all "user added" devices with the "show device userAdded" command.

add ldap

Adds new Lightweight Directory Access Protocol (LDAP) group ID and assigns it a role name. (Tech or Admin)

Syntax

`add ldap ldapGroudID` *`id`* `mapto rolename` *`name`*

Parameters

`id`

Type: INTEGER

Group ID number for the LDAP (Range from 1 to 65535)

`name`

Type: STRING

Role Name

Example

```
add ldap ldapGroupId 4569 mapto roleName admin
Success

add ldap ldapGroupId 6596 mapto roleName Tech
Success
```

Related Commands

`delete ldap`
`set ldap mode`
`show ldap server`

add snmp

Creates an new SNMP user or trap server. (Please see Section 5 of this manual for additional details on SNMP support)

Syntax

```
add snmp arg name
```

Parameters

arg

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`trapServer vc2Trap ipAddress <address> community <comm>`	Add new trap server at the specified IP Address
`user v2c accessLevel readOnly community`	Add new SNMP user
`user v3 accessLevel readOnly auth MD5 encrypted no username <name> password <password>`	Add new SNMP user

name

Type: STRING

IP address of trapserver or name of new SNMP user (Password must be 8 to 127 characters)

Example

```
add snmp user v3 accessLevel readOnly auth MD5 encrypted no
username john password abc12345
Success

add snmp trapServer v2cTrap ipAddress 192.168.0.231 community john
Success
```

Related Commands

delete snmp
show snmp

add snmp netNode (v3)

Adds new SNMP netNode for use with the ZeeVee Visualization, Analysis and Monitoring tool (VAM).

Syntax

```
add snmp netnode ipAddress ipaddr snmp v3 authType auth username
name password pass
```

Parameters

ipaddr

Type: STRING

IP Address of the new netNode (IP Address of switch on the network)

auth

Type: STRING

argument	Description
sha1	Use SHA-1 Hash
sha512	Use SHA-512 Hash

name

Type: STRING

Username

pass

Type: STRING

Password.

Example

```
add snmp netNode ipAddress 192.168.0.1 snmp v3 authType sha512
username admin password netgear96
Success
```

Related Commands

```
delete snmp netNode
set snmp netNode
show snmpNode
```

add snmp netNode (v2c)

Adds new SNMP netNode for use with the Visualization, Analysis and Monitoring tool (VAM).

Syntax

```
add snmp netnode ipAddress ipaddr snmp v2c communityName name
```

Parameters

ipaddr

Type: STRING

IP Address of the new netNode (IP Address of switch on the network)

name

Type: STRING

Community Name

Example

```
add snmp netNode ipAddress 192.168.0.55 snmp v2c communityName home
Success
```

Related Commands

```
delete snmp netNode
set snmp netNode
show snmpNode
```

add zoneDisplay

Adds a display or video-wall to an existing zone.

Care should be taken that individual displays found within walls are not added to a Zone. This would result in the same display being in a zone more than once.

Syntax

```
add zoneDisplay name id
```

Parameters

name

Type: STRING

The name of the zone. Names are case-sensitive. ("All" is an option to add selected id to every current zone)

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Example

```
add zoneDisplay Zone1 Decoder5
Success

add zoneDisplay All Decoder2
Success
```

Related Commands

```
create zone
delete zone
delete zoneDisplay
show zones
```

authenticate username

Used by browsers to authenticate users accessing ZMP.

Note: This command is not intended to be run directly from the API command line interface.

Syntax

`authenticate username` *`user`* `password` *`pwd`* `token` *`tkn`* `newPasword` *`npwd`*

Parameters

`user`

Type: STRING

The name of the user. Names are case-sensitive.

`pwd`

Type: STRING

Password. Passwords are case-sensitive

`tkn`

Type: STRING

Token.

`npwd`

Type: STRING

New Password.

channel

Will cycle through all encoders (of the same type as the decoder) that have a number (channel) suffix, "_nnn", where nnn is an integer (channel).

If there are encoders with names: enc_1, enc_100, enc_50, then a decoder will cycle through them in the order: enc_1, enc_50, enc_100, then back to enc_1.
If there are no encoders (of the same type as the decoder) with the channel suffix, an error is returned.

Only fastSwitch connection types is supported. If there was already a connection of some other type, it is changed to fastSwitched.

If the decoder has no connection, the encoder with the lowest channel suffix will be connected using fast-switch.

If the decoder has a connection to an encoder that does not have the channel suffix, then it will connect to the encoder that has the lowest channel suffix.

Note: In fastSwitch mode the join videoSource <decoder> command must be used to set audio to follow video join. Otherwise audio will not follow the video during channel up/ down command.

Syntax

`channel` *`direction`* `<decoder-id>`

Parameters

`direction`
Type: STRING

argument	Description
`up`	cycle to next higher numbered encoder
`down`	cycle to next lower numbered encoder

`decoder-id`
Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

Example

```
channel up MyDecoder
Channel changed to Channel_2
Success
```

Related Commands

`join videoSource`

clone Multiview

Used to create a copy of an exisitng multiview. (ZyPer4K only) Once created, the new multiview will be listed under the Multiview menu within the built-in ZMP.

Use the `set multiview` command to set a source encoder to a specified location and size within the multiview.

Refer to Creating a Multiview Screen (page 46) for information on managing multiview displays in the built-in ZMP.

Syntax

```
clone multiview name to newmvname
```

Parameters

name

Type: STRING

The name of the existing multiview to be cloned. Names are case-sensitive.

newmvname

Type: STRING

The name of the new multiview. The name of the multiview cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
clone multiview mv2x2 to newmv2x2
Success
```

Related Commands

```
delete multiview
delete multiviewWindow
set multiview
set multiview audioSource windowNumber
show multiviews config
show multiviews status
```

convert PID 12G SDI Devices

Used to update the Product Identification Code for ZyPer4K 12G SDI Encoders. (ZyPer4K only).

This command typically woudl only be run once and only under the direction of the Kramer/ZeeVee support team. PID update file to be obtained from support team.

Syntax

```
convertPID ZyPer4K sdi12g toNewPid|toOldPid filename
```

Parameters

Pid Direction

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`toNewPid`	Update all 12G SDI encoder to new Product Identification Code
`toOldPid`	Revert all 12 SDI encoders to old Product Identification Code

filename

Type: STRING

The name of the file located on the ZMP to setting new/old Product Identification Code. File to be obtained from Kramer/ZeeVee support team.

Example

```
convertPid Zyper4K sdi12g toNewPid semtech_blueriver_4.1.2.9_
build3943-zeevee-PLUS_12GSDI.apz
Success
```

create account

Creates a new user account with assigned password.

Syntax

```
create account name passwordOption
```

Parameters

name

Type: STRING

The name of the account The name of the account cannot exceed 255 characters in length. Names are case-sensitive.

passwordOption

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`password`	Enter the password to be used by this account.
`tempInitialPassword`	System will generate a temporary one time use password that will expire after the first use.

Examples

```
create account bartender tempInitialPassword
result: password=ovmOH;arZasuHS, expires=immediate
Success

create account bartender password 12345
Success
```

Related Commands

`delete account`

create kvm

Creates a new kvm in the database on the Management Platform. (ZyPer4K-XS/XSE/XR only)

Syntax

```
create kvm name
```

Parameters

name

Type: STRING

The name of the kvm. Names are case-sensitive.

Example

```
create kvm ControlRoom
Success
```

Related Commands

```
delete kvm
rename kvm
set kvm
show kvm
start kvm
stop kvm
```

create Multiview

Creates an empty multiview display. (ZyPer4K only) Once created, the new multiview will be listed under the Multiview menu within the built-in ZMP.

Use the `set multiview` command to set a source encoder to a specified location and size within the multiview.

Refer to Creating a Multiview Screen (page 46) for information on managing multiview displays in the built-in ZMP.

Syntax

```
create multiview name
```

Parameters

name

Type: STRING

The name of the multiview. The name of the multiview cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
create multiview myMultiview
Success
```

Related Commands

```
delete multiview
delete multiviewWindow
set multiview
set multiview audioSource windowNumber
show multiviews config
show multiviews status
```

create presetNew

Creates a new preset. Once created, the new preset will be listed under the Preset menu within the built-in ZMP.

Syntax

`create presetNew` *`name`* `commands` *`connections`*

Parameters

`name`

Type: STRING

The name of the preset. The name of the preset cannot exceed 255 characters in length. Names are case-sensitive.

`connections`

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`empty`	This preset has no commands or connections associated with it.
`existingConnections`	Use the current connections to generate the preset

Example

```
create presetNew EveningShutDown commands existingConnections
Success
```

Related Commands

`create presetSchedule`
`delete preset`
`run preset`
`set preset`
`show preset`

create presetSchedule

Inserts an existing preset into the schedule calendar Once created, the item must be assigned months/days/time to execute. By default without further setting, the preset will be schedule to occur every hour of every day.

Use the `set preset zoneDisplay` command to assign description, commands and schedule to the new schedule.

Syntax

`create presetSchedule` *`presetname`* `schedule` *`name`*

Parameters

`presetname`

Type: STRING

The name of an existing preset.

`name`

Type: STRING

The name of the schedule. The name of the schedule cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
create presetSchedule EveningShutDown schedule GoHome
Success
```

Related Commands

`create presetNew`
`delete preset`
`run preset`
`set preset`
`show preset`

create role

Creates a account role with specified access level.

Syntax

```
create role name allSubsystems maxAccess accessLevel
```

Parameters

name

Type: STRING

The name of the role The name of the role cannot exceed 255 characters in length. Names are case-sensitive.

accessLevel

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`admin`	Role has full administration privileges. No restrictions.
`config`	Role is able to configure existing items (endpoints, multiview, walls etc..) but cannot create/delete items.
`join`	Can only issue join commands
`none`	No access. Can only view the Help tab.
`view`	Role can only view. Cannot perform any actions.

Examples

```
create account bartender tempInitialPassword
result: password=ovmOH;arZasuHS, expires=immediate
Success

create account bartender password 12345
Success
```

Related Commands

`delete role`

create videoWall

Creates an empty 2x2 video wall. Once created, the new video wall will be listed under the Display Config menu within the built-in ZMP.

Use the `join videoWall`command to assign a source encoder to the wall. To modify the size of the video wall and/or control bezel parameters, use the `set videoWall` command.

Refer to Creating Video Walls (page 41) for information on managing video walls in the ZMP.

Syntax

```
create videoWall name
```

Parameters

name

Type: STRING

The name of the video wall. The name of the video wall cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
create videoWall myWall
Success
```

Related Commands

`delete videoWall`
`set videoWall size`

create zone

Creates an empty zone Once created, the new zone will be listed under the Zones menu within the built-in ZMP.

Use the `add zoneDisplay` command to assign decoders or video walls to the zone.

Syntax

`create zone` *`name`*

Parameters

`name`

Type: STRING

The name of the zone. The name of the zone cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
create zone Zone1
Success
```

Related Commands

`add zoneDisplay`
`delete zone`
`delete zoneDisplay`
`show zones`

dataConnect

Connects two devices for IR or RS232 communication over a specified TCP port. (Note TCP port only valid for connection between device and server. Not valid for connection between 2 devices)

The feature of dataConnect was added to allow a third party to connect to the ZMP server with a specific port and pass raw or telnet API commands (depending on the mode) to the server and port which is designated for a particular encoder or decoder.

Syntax

`dataConnect` *`id1 id2 mode`* `tunnelPort` *`port`*

Parameters

`id1`

Type: STRING

The name of the first device. String names are case-sensitive.

`id2`

Type: STRING

The name of the second device or server. String names are case-sensitive. Note: Connection to server only supported for ZyPer4K. ZyPerUHD and ZyPer4K-XS/XR are not supported for this function.

`mode`

Type: STRING

ir or rs232

`port`

Type: INTEGER

TCP-Port #. Integer range from 1,024 to 49,152

Example

```
dataConnect MediaPlayer server rs232 tunnelPort 2345
tunnel TCP port = 2345; telnet handshake mode
Success
```

Related Commands

`show dataTunnels`
`set server dataTunnelMode`

Notes on Tunnel Ports

There is a very convenient way to get RS232 data: TUNNELS.

```
Zyper$ dataConnect Decoder_1 server rs232
Dynamically assigned tunnel TCP port = 4100; telnet handshake mode
Success
Zyper$
Zyper$ show dataTunnels
data-sessions(d8:80:39:9b:9:a2);
  device: name=Decoder_1
  rs232Tunnel: port=4100
  rs232Tunnel-connections: none
Success
Zyper$
```

You can then connect to that tunnel port using TCP. Whatever is sent is forwarded to the device. Whatever the device returns is received on that TCP connection.

In the easiest case, you can just use telnet to connect to the tunnel.

```
You can specify the port number as well:
Zyper$ dataConnect Decoder_1 server rs232 tunnelPort 4101
tunnel TCP port = 4101; telnet handshake mode
Success
Zyper$
```

You can set the default TCP connection mode: raw|telnet (defaults to telnet).

```
Zyper$ set server dataTunnelMode raw|telnet
```

When in telnet mode the IAC commands are sent/received. Although most telnet clients will also work fine in raw mode.

delete account

Deletes the specified account from the Management Platform database.

Syntax

```
delete account id
```

Parameters

id

Type: STRING

The name of the account. String names are case-sensitive.

Example

```
delete account bartender1
Success
```

Related Commands

`create account`

delete allConfiguration

Deletes all device and server information from the Management Platform. The network configuration is preserved.

Syntax

```
delete allConfiguration action
```

Parameters

action

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`reboot`	Unit is automatically rebooted
`restart`	The ZyPer server service is restarted
`shutdown`	Unit is shutdown

Example

```
delete allConfiguration restart
delete allConfiguration reboot
```

Related Commands

`factoryDefaults device`

delete device

Deletes the specified device from the Management Platform database.

Note that if the deleted device remains on the network, then it will be rediscovered by the Management Platform and reposted to the database. (Unless in Isolaiton Mode) To permanently remove a device from the database, physically disconnected it and execute the `delete device` command.

Syntax

```
delete device id
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Example

```
delete device myDevice
Success

delete device 0:1e:c0:f6:42:a1
Success
```

Related Commands

`factoryDefaults device`

delete kvm

Deletes the specified kvm from the database on the Management Platform. (ZyPer4K-XS/XSE/XR only)

Can also be used to delete a specific location from a configured KVM using Row/Column location.

Syntax

```
delete kvm name

delete kvm name row <int> col <int>
```

Parameters

name

Type: STRING

The name of the kvm. Names are case-sensitive.

Example

```
delete kvm ControlRoom
Success

delete kvm ControlRoom row 2 col 1
Success
```

Related Commands

```
create kvm
delete kvm
rename kvm
show kvm
start kvm
stop kvm
```

delete ldap

Deletes the specified Lightweight Directory Access Protocol (LDAP) from the database on the Management Platform.

Syntax

```
delete ldap ldapGroupID id
```

Parameters

id

Type: INTEGER

LDAP Group ID of the LDAP to be deleted

Example

```
delete ldap ldapGroupId 6596
Success
```

Related Commands

add ldap
set ldap
show ldap
s

delete multiview

Deletes the specified multiview from the database on the Management Platform.
(ZyPer4K family only)

Syntax

```
delete multiview name
```

Parameters

name

Type: STRING

The name of the multiview. Names are case-sensitive.

Example

```
delete multiview myMultiview
Success
```

Related Commands

```
create multiview
delete multiviewWindow
set multiview
set multiview audioSource windowNumber
show multiviews config
show multiviews status
```

delete multiviewWindow

Deletes the specified window from an existing multiview. (ZyPer4K family only)

Syntax

```
delete multiviewWindow name window arg
```

Parameters

name

Type: STRING

The name of the multiview. Names are case-sensitive.

arg

Type: INTEGER

Window number to remove. Integer range from 1 to 9

Example

```
delete multiviewWindow myMultiview window 5
Success
```

Related Commands

```
create multiview
delete multiview
set multiview
set multiview audioSource windowNumber
show multiviews config
show multiviews status
```

delete preset

Deletes the specified preset, preset runlog or preset schedule from the system.

Note: Runlog is history of when the preset has been executed. Deleting the runlog does not impact the preset itself or the schedule.

Syntax

```
delete preset name
delete preset name runLog
delete preset name schedule schname
```

Parameters

name

Type: STRING

The name of the preset. Names are case-sensitive.

schname

Type: STRING

The name of the preset schedule. Names are case-sensitive.

Examples

```
delete preset lunch runLog
Success

delete preset lunch schedule eat
Success
```

Related Commands

create preset
run preset
set preset
show preset

delete role

Deletes the specified role from the Management Platform database.

Syntax

```
delete role id
```

Parameters

id

Type: STRING

The name of the role. String names are case-sensitive.

Example

```
delete role bartender
Success
```

Related Commands

`create role`

delete snmp

Deletes an existing SNMP user or trap server. (Please see Section 5 of this manual for additional details on SNMP support)

Syntax

```
delete snmp arg name
```

Parameters

arg

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`trapServer vc2Trap ipAddress <address> community <comm>`	Delete trap server at the specified IP Address
`user v2c`	Delete SNMP user
`user v3`	Delete SNMP user

name

Type: STRING

IP address and community of trapserver or name of new SNMP user

Example

```
delete snmp trapServer v2cTrap 192.168.0.231 community john
Success

delete snmp user v3 username john
Success
```

Related Commands

add snmp
show snmp

delete snmp netNode

Deletes the specified netNode associated with the ZeeVee Visualization, Analysis, Monitoring tool (VAM)

Syntax

```
delete snmp netNode arg ident
```

Parameters

arg

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
byName	Name of the Netnode
byId	ID of the Netnode

ident

Type: STRING or INTEGER

Name of the Netnode for the numeric ID of the Netnode

Example

```
delete snmp netNode byId 3
Success

show snmp netNode all general
netNode(1);
  netNode.gen; name="myZMPserver", chassisId=54:b2:03:f0:ee:99,
type=NA, portErrors=0, ip=192.168.0.21, state=up,
uptime=0d:1h:4m:9s
  netNode.gen; make=zeevee, descr="ZyPer Management Platform",
version=" 3.3.39589", location="Wiring closet"
netNode(2);
  netNode.gen; name="NA", chassisId=8c:3b:ad:68:6b:34,
type=bridge+router, portErrors=1, ip=192.168.0.1, state=up,
uptime=3d:4h:31m:12s
  netNode.gen; make=netgear, descr="M4300-8X8F ProSAFE 8-port
10GBASE-T and 8-port 10G SFP+", version=" 12.0.17.13",
location="NA"
Success
```

Related Commands

add netNode
show netNode

delete videoWall

Deletes the specified video wall from the database on the Management Platform.

Syntax

```
delete videoWall name
```

Parameters

name

Type: STRING

The name of the video wall. Names are case-sensitive.

Example

```
delete videoWall myWall
Success
```

Related Commands

```
create videoWall
set videoWall size
```

delete zone

Deletes the specified zone from the database on the Management Platform.

Syntax

```
delete zone name
```

Parameters

name

Type: STRING

The name of the zone. Names are case-sensitive.

Example

```
delete zone zone1
Success
```

Related Commands

add zoneDisplay
create zone
delete zoneDisplay
show zones

delete zoneDisplay

Deletes the specified display from an existing zone.

Syntax

```
delete zoneDisplay name id
```

Parameters

name

Type: STRING

The name of the zone. Names are case-sensitive.

id

Type: STRING or MAC Address

The name or MAC address of the decoder/display. String names are case-sensitive.

Example

```
delete zoneDisplay myzone mydisplay1
Success
```

Related Commands

```
add zoneDisplay
create zone
delete zone
show zones
```

diagnostics device

Runs a set to test diagnostics on the specified device

Syntax

```
diagnostics device id
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Possible Results

Decoder/Encoder:

error, Device is down
warning, Device has no HDMI link
warning, Device rebooted %d times in the last minute
warning, Device rebooted %d times in the last hour
warning, Device rebooted %d times in the last day

Decoder:

error, HDMI Audio stream connection without video connection.
warning, HDCP is forced on, but may not be supported by display device (however it is unlikely)
error, Decoder has never received a valid EDID
warning, Decoder resolution exceeds display EDID maximum -- very likely this will not work
warning, Decoder using encoder resolution, which may not be display's preferred based on its EDID
warning, Decoder using encoder resolution AND ignoring display EDID, which may allow resolution to exceed display capability
warning, Decoder using user-defined resolution, which may allow resolution to exceed display capability
error, Encoder down
warning, Encoder hdmi down
error, Encoder has multiview conflict with genlock
warning, Encoder stream disabled
warning, Video stream interrupted %d times in the last minute, indicating likely network problem
warning, Video stream interrupted %d times in the last hour, indicating likely network problem
warning, Video stream interrupted %d times in the last day, indicating possible network problem
warning, Encoder and decoder fps are not equal -- will result in very bad video
warning, Encoder and decoder fps are not equal -- will result in very bad video
warning, Encoder and decoder fps are not equal, but multiple of 2; this may still produce bad video
warning, Encoder and decoder HDCP versions are not the same
info, Encoder HDCP is disabled; this will prevent copyrighted material from display
info, Encoder HDCP is set to version 1.4; this may prevent copyrighted material from display

Encoder:

info, HDCP is disabled; this will prevent copyrighted material from display
info, HDCP is set to version 1.4; this may prevent copyrighted material from display

Examples

```
diagnostics device Top-Right
device(d8:80:39:9a:7f:ec);
  device.diags.summary; status=complete, error=0, warning=0, info=0
Success

diagnostics device ABC
device(d8:80:39:9a:96:7);
  device.diags.info.1; message=HDCP is disabled; this will prevent
copyrighted material from display
  device.diags.summary; status=complete, error=0, warning=0, info=1
Success

diagnostics device encoder1
device(34:1b:22:80:26:2a);
  device.diags.warning.1; message=Device has no HDMI link
  device.diags.summary; status=complete, error=0, warning=1, info=0
Success

diagnostics device MyEncoder
device(34:1b:22:80:63:9c);
  device.diags.error.1; message=Device is down
  device.diags.summary; status=complete, error=1, warning=0, info=0
Success


diagnostics device MeetingRoom6
device(34:1b:22:80:57:7d);
  device.diags.error.1; message=Device is down
  device.diags.info.1; message=No video connection
  device.diags.warning.1; message=HDCP is forced on, but may not be
supported by display device (however it is unlikely)
  device.diags.error.2; message=Decoder has never received a valid
EDID
  device.diags.summary; status=complete, error=2, warning=1, info=1
Success
```

dumpusb

Outputs details about USB devices found in ZyPerUHD, ZyPerUHD60 and/or ZyPer4K units. Information includes MAC address and ICRON IP_address if ICRON USB found in ZyPer4K unit.

Syntax

```
dumpusb
```

Example

```
dumpusb
Encoders/Decoders usb reported mac
  device UHDdec(UHDdec), usb mac 34:1b:22:80:57:df
  device UHDenc2(UHDenc2), usb mac 34:1b:22:80:7f:3d
  device Z4Kdec1(Z4Kdec1), usb mac 0:1b:13:1:1f:79
  device Arts_Encoder_1(Arts_Encoder_1), usb mac 0:1b:13:1:1e:90
Icrons reported info
  owner Arts_Encoder_1(80:1f:12:4d:9b:6b), deviceType local,
icronMac 0:1b:13:1:1e:90, ipAddr 169.254.4.123, fwRev
1.9.4, pairedInfoRcvd yes, numPairedMacs 1, 0:1b:13:1:1f:79
Z4Kdec1(Z4Kdec1)
  owner Z4Kdec1(80:1f:12:4d:2c:ff), deviceType remote, icronMac
0:1b:13:1:1f:79, ipAddr 169.254.4.125, fwRev 1.9.4, pairedInfoRcvd
yes, numPairedMacs 1, 0:1b:13:1:1e:90 Arts_Encoder_1(Arts_
Encoder_1)
Success
```

Note: This is a hidden command and will not appear in HELP

events

Causes the events mode to be entered.

Syntax

```
events
```

Server sends initial events and new events as they occur to the telnet session. Any character entered to the server causes the mode to exit back to the API prompt.

See Section 4 of this document for additional details on the events feature.

factoryDefaults device

Set the specified device to the factory-default settings.

Syntax

```
factoryDefaults device id
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Example

```
factoryDefaults device Airshow
Success

factoryDefaults device 0:1e:c0:f6:a8:c3
Success
```

Related Commands

`delete allConfiguration`

flashLeds

Physically identifies the specified device on the network. When this command is executed, the LED indicators on the device will flash for 5 seconds.

Syntax

```
flashLeds id
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Example

```
flashLeds myEncoder
Success

flashLeds 0:1e:c0:f6:59:13
Success
```

generate tls ca privKeyPass

Used to generate a local Transport Layer Security Certificate Authority private key.

Syntax

```
generate tls ca privKeyPass privKey country country state state
locality local organization org organizationUnit orgunit email
email

Enter passphrase: passphrase
```

Parameters

privKey
Type: STRING | *

Private key phrase. String. If * used; will be prompted for passphrase at the end.

country
Type: STRING

2 character string representing Country. Example "US"

state
Type: STRING

2 character string representing State. Example "MA"

local
Type: STRING

String representing local town/city. Example "Billerica"

org
Type: STRING

String representing organization. Example "ZeeVee"

orgunit
Type: STRING

String representing organization units. Example "money"

email
Type: STRING

String representing email address. Example "aweeks@zeevee.com"

passphrase

Type: STRING

Private phrase used in generation of the tls key. Prompted if * used earlier in command.

Example

```
generate tls ca privKeyPass * country US state MA locality
Billerica organization ZeeVee organizationUnit money email aweeks@
zeevee.com
Enter passphrase: ******
Success
```

Related Commands

```
generate tls server csr privKeyPass
show tls pem ca privKey

show tls pem ca privKey
pemData:
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,ADB163FA01562B533B617FA5792AB7F1

........

-----END RSA PRIVATE KEY-----
Success
```

generate tls device csr privKeyPass

Used to generate a local Transport Layer Security server Certificate Signing Request private key for a device.

Syntax

```
generate tls device csr privKeyPass privKey fqdn domain
country country state state locality local organization org
organizationUnit orgunit email email

Enter passphrase: passphrase
```

Parameters

privKey

Type: STRING | *

Private key phrase. String. If * used; will be prompted for passphrase at the end.

domain

Type: STRING

String representing fully qualified domain name. Example "zeevee.com"

country

Type: STRING

2 character string representing Country. Example "US"

state

Type: STRING

2 character string representing State. Example "MA"

local

Type: STRING

String representing local town/city. Example "Billerica"

org

Type: STRING

String representing organization. Example "ZeeVee"

orgunit

Type: STRING

String representing organization units. Example "money"

email

Type: STRING

String representing email address. Example "aweeks@zeevee.com"

passphrase

Type: STRING

Private phrase used in generation of the tls key. Prompted if * used earlier in command.

Example

```
generate tls device csr privKeyPass * fqdn zeevee.com country US
state MA locality Billerica organization ZeeVee organizationUnit
money email aweeks@zeevee.com
Enter passphrase: ******
Success
```

Related Commands

```
generate tls ca privKeyPass
show tls pem server csr
show tls pem server privKey

show tls pem server csr
pemData:
-----BEGIN CERTIFICATE REQUEST-----
................
-----END CERTIFICATE REQUEST-----
Success
```

generate tls server csr privKeyPass

Used to generate a local Transport Layer Security server Certificate Signing Request private key.

Syntax

```
generate tls server csr privKeyPass privKey fqdn domain
country country state state locality local organization org
organizationUnit orgunit email email

Enter passphrase: passphrase
```

Parameters

privKey

Type: STRING | *

Private key phrase. String. If * used; will be prompted for passphrase at the end.

domain

Type: STRING

String representing fully qualified domain name. Example "zeevee.com"

country

Type: STRING

2 character string representing Country. Example "US"

state

Type: STRING

2 character string representing State. Example "MA"

local

Type: STRING

String representing local town/city. Example "Billerica"

org

Type: STRING

String representing organization. Example "ZeeVee"

orgunit

Type: STRING

String representing organization units. Example "money"

email

Type: STRING

String representing email address. Example "aweeks@zeevee.com"

passphrase

Type: STRING

Private phrase used in generation of the tls key. Prompted if * used earlier in command.

Example

```
generate tls server csr privKeyPass * fqdn zeevee.com country US
state MA locality Billerica organization ZeeVee organizationUnit
money email aweeks@zeevee.com
Enter passphrase: ******
Success
```

Related Commands

```
generate tls ca privKeyPass
show tls pem server csr
show tls pem server privKey

show tls pem server csr
pemData:
-----BEGIN CERTIFICATE REQUEST-----
.................
-----END CERTIFICATE REQUEST-----
Success
```

generate tls radius csr privKeyPass

Used to generate a local Transport Layer Security server Certificate Signing Request private key.

Syntax

```
generate tls radius csr privKeyPass privKey fqdn domain
country country state state locality local organization org
organizationUnit orgunit email email

Enter passphrase: passphrase
```

Parameters

privKey

Type: STRING | *

Private key phrase. String. If * used; will be prompted for passphrase at the end.

domain

Type: STRING

String representing fully qualified domain name. Example "zeevee.com"

country

Type: STRING

2 character string representing Country. Example "US"

state

Type: STRING

2 character string representing State. Example "MA"

local

Type: STRING

String representing local town/city. Example "Billerica"

org

Type: STRING

String representing organization. Example "ZeeVee"

orgunit

Type: STRING

String representing organization units. Example "money"

email

Type: STRING

String representing email address. Example "aweeks@zeevee.com"

passphrase

Type: STRING

Private phrase used in generation of the tls key. Prompted if * used earlier in command.

Example

```
generate tls radius csr privKeyPass * fqdn zeevee.com country US
state MA locality Billerica organization ZeeVee organizationUnit
money email aweeks@zeevee.com
Enter passphrase: ******
Success
```

Related Commands

```
generate tls ca privKeyPass
show tls pem server csr
show tls pem server privKey

show tls pem server csr
pemData:
-----BEGIN CERTIFICATE REQUEST-----
................
-----END CERTIFICATE REQUEST-----
Success
```

help

Provides a listing of API commands grouped or sorted in various ways.

Syntax Options

```
help
    help all alphabetical
    help all byConcept
    help all bySubsystem
    help all byAccessLevel
    help concept <helpConcepts>
    help subsystem <helpSubsystems>
    help accessLevel <helpAccessLevels>
    help search string <keyWord:string>
```

Example

```
help

 Help commands:
    help
    help all alphabetical
    help all byConcept
    help all bySubsystem
    help all byAccessLevel
    help concept <helpConcepts>
    help subsystem <helpSubsystems>
    help accessLevel <helpAccessLevels>
    help search string <keyWord:string>
    <command> help
    <command> ?
    ?

** NOTE: Use <tab> to complete a command **

Success
```

join

Joins the specified decoder (display) with the specified encoder (source). The `mode` parameter must be specified and defines the type of join to execute.

- analogAudio
  Embeds analog audio stream from the encoder on the output of the decoder. The audio is from the (analog) Audio jack on the encoder. Will force UHD60 Dante enabled encoder into Dante Transmit mode.
  In order to control what type of audio is being output from the decoder, refer to the `set decoder analogAudioOut` command.

- danteAudio
  Will force UHD60 Dante enabled encoder into Dante Receive mode. This allows a received Dante audio stream to be transmitted as regular UHD60 audio to any UHD60 decoder.

- fastSwitched
  Allows the joining of an encoder and decoder with no video dropout. In order to make use of this feature, the resolution and frame rate of the "new" encoder must be the same as the previous encoder.

- genlocked
  This mode provides a very low-latency, all-purpose method of joining an encoder and decoder. (ZyPer4K family only)

- genlockedScaled
  This mode provides a very low-latency, all-purpose method of joining an encoder and decoder that includes scaling up or down at the decoder/display.

- hdmiAudio
  Embeds hdmi-downmix audio from an encoder to specified decoder.

- multiview
  Join the configured multiview to a display (decoder) (ZyPer4K family only)

- video
  Joins video only from encoder to decoder. No audio.

- videoWall
  Join the encoder to the named video-wall

- window
  Join any portion of a source to any portion of a display

- usb
  Creates USB connection between encoder and decoder. Note that multiple connections are valid.

- none
  Special command to disconnect existing connections (joins) Example: join none decoder fastSwitched

Syntax

```
join enc dec mode
join none dec fastSwitched
```

Parameters

enc

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

dec

Type: STRING or MAC Address

The name or MAC address of the decoder. Can also be name of existing video-wall String names are case-sensitive.

zone

Type: STRING

The name of an existing zone. String names are case-sensitive.

mode

Type: STRING

Supply one of the following arguments before executing this command.

argument	Description
`analogAudio`	Embed audio from the specified encoder. Force Dante enabled UHD60 encoder into Dante transmit mode.
`danteAudio`	Force UHD60 Dante enabled encoder into Dante receive mode. (Z4K-XS units will be both send and receive)
`fastSwitched`	Join in "fast-switched" mode
`genlocked`	Low-latency join mode (ZyPer4K family only)
`genlockedScaled`	Low-latency with scale up/down (ZyPer4K family only)
`hdmiAudio`	Join hdmi-audio to either hdmi-out or analog-out. Note this command will cause hdmiAudioFollowVideo=False for specified decoder. See join videoSource command on next page.
`multiview`	Join a multiview to a display (ZyPer4K family only)
`videoWall`	Join a source to a video-wall
`video`	Join video only (audio not joined)
`window`	Join any portion of a source to any portion of a display (ZyPer4k family only)
`usb`	Establish USB connection
`"none"`	Disconnect existing joins

Notes:
Multiviews cannot be joined to a zone.
USB cannot be joined to a zone.

Examples

```
join myEncoder1 myDecoder2 fastSwitched
Success

join myEncoder1 myDecoder2 hdmiAudio
Success
(Note: If Dante Encoder currently set to Receive mode, this will set the
Encoder to Dante Transmit mode and force a reboot of the encoder)

join myMultiview2 Display4 multiview
Success

join myEncoder1 myWall videoWall
Success

join none myDecoder1 fastSwitched

join myEncoder1 myDecoder2 danteAudio
(Sets myEncoder1 to Dante Receive mode.  Will force reboot of UHD60 encoder
if not already in Dante Receive mode)
```

Window Example

```
join myEncoder1 myDecoder2 window viewportSource 0 0 1920 1080
viewportDest 500 500 500 500

(ViewportSource parameters are starting X/Y coordinates of the
source and desired X/Y size)

(ViewportDest parameters are starting X/Y coordinates in the
display and desired X/Y size)
```

join videoSource

Tells a decoder to automatically join corresponding audio from a source encoder whenever a join command is used to join video.

Syntax

```
join videoSource dec mode
```

Parameters

dec

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

mode

argument	Description
audio	automatically join audio from connected encoder (ZyPerUHD only)
hdmiAudio	automatically join hdmi-audio from connected encoder (ZyPer4K family only)

Example

```
join videoSource MyDecoder hdmiAudio
Success
```

Related Commands

join hdmiAudio

load account

Uploads text and/or images to be displayed prior to and after the login screen. Can be used a warning or any other purpose.

Syntax

```
load account all PrePost Arg file
```

Parameters

PrePost

argument	Description
preLoginBanner	Specified text or image will appear before login
postLoginBanner	Specified text or image will appear after login

Arg

argument	Description
terminal	Provides pre or post login message when using SSH to access the API command line interface. Note: This feature does not work when using Telnet to access the API command line interface.
webText	Text that will appear before/after ZMP-GUI login
webImage	Image that will appear before/after ZMP-GUI login

file

Type: STRING

The name of the file to load. Text or .PNG

Examples

```
load account all preLoginBanner webImage DOD-Seal.png
Success

load account all preLoginBanner webText securePre.txt
Success

load account all postLoginBanner webImage mickey.png
Success
```

load encoderEdid

Uploads an EDID file to the specified encoder.

Important Note: Auto-EDID mode should be disabled when loading a specific EDID to an encoder. Otherwise the loaded EDID will immediately get replaced by the Auto-EDID option.

Syntax

`load encoderEdid` *`enc mode file`*

Parameters

`enc`

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

`mode`

argument	Description
`auto`	use whatever EDID information is provided by connected decoder
`builtIn`	use one of the EDID files provided by ZeeVee. Many options available covering various 4k settings. See list below.
`default`	use default EDID with maximum capabilities of the encoder
`saved`	use a file that user has previously saved to the system with the save device-edid command

`file`

Type: STRING

The name of the file to load.

Built in EDID options

zyper-default
zyper4k25
zyper4k30

Built in EDID options continued

zyper4k50
zyper4k50-420
zyper4k50-420_hdmi14
zyper4k50-hbraudio
zyper4k50-hd-hdr
zyper4k50-hdr
zyper4k50-hdr-bt2020
zyper4k50-hdr-bt2020-hbraudio
zyper4k50-hdr-hbraudio
zyper4k60
zyper4k60-420
zyper4k60-420_hdmi14
zyper4k60-hbraudio
zyper4k60-hd-hdr
zyper4k60-hdr
zyper4k60-hdr-bt2020
zyper4k60-hdr-bt2020-hbraudio
zyper4k60-hdr-hbraudio
zyperHd50
zyperHd60
zyperPc
zyperUhd25
zyperUhd25-hbraudio
zyperUhd30
zyperUhd30-hbraudio
zyperUhd50
zyperUhd50-420
zyperUhd50-420_hdmi14
zyperUhd50-hbraudio
zyperUhd50-hd-hdr
zyperUhd50-hdr
zyperUhd50-hdr-bt2020
zyperUhd50-hdr-bt2020-hbraudio
zyperUhd50-hdr-hbraudio
zyperUhd60
zyperUhd60-420
zyperUhd60-420_hdmi14
zyperUhd60-hbraudio
zyperUhd60-hd-hdr
zyperUhd60-hdr
zyperUhd60-hdr-bt2020
zyperUhd60-hdr-bt2020-hbraudio
zyperUhd60-hdr-hbraudio

Examples

```
load encoderEdid myEncoder saved myEDID.bin
Success

load encoderEdid myEncoder builtIn zyper4k60
Success
```

Related Commands

```
save deviceEdid
set server autoEdidMode
```

load idleImage

Uploads an image to use at ZyPerUHD background when no video source streamed to the decoder.

Syntax

`load idleImage dec filename file`

Parameters

dec

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

file

Type: STRING

The name of the file to load. (Must already exist on ZMP in Files directory)

Examples

```
load idleImage myDecoder filename background.jpg
Success
```

Notes:

```
Image must be in .JPG format
Image must be 1280 x 720 in size
(Will output from decoder at this resolution)

Image file must be previously copied onto ZMP into the Files
directory using FTP.  Alternately file can be loaded via the GUI.
See Display Grid "Config" tab.
```

Related Commands

`save deviceEdid`

load tls ca cert

Options for loading Transport Layer Security Certifiate Authority certification

Syntax

```
load tls ca cert fromInput *
```

```
load tls ca cert fromFile filename
```

Parameters

input

Type: STRING

String representing the Certficiate Authority. The system will prompt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls ca cert fromInput *
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE-----
MIIF1TCCA72gAwIBAgIBADANBgkqhkiG9w0BAQsFADB9MRswGQYJKoZIhvcNAQkB
..............RDz+0llBNWe2
-----END CERTIFICATE-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls ca privateKey privKeyPass
show tls summary

load tls ca privateKey

Options for loading Transport Layer Security Certifiate Authority Private Key

Syntax

```
load tls ca privateKey privKeyPass * fromInput *
```

```
load tls ca privateKey privKeyPass * fromFile filename
```

Parameters

input

Type: STRING

String representing the Private Key. The system will promt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls ca privateKey privKeyPass * fromInput *
Enter passphrase: ******
Enter PEM text (ctr-d to end):
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,16DD663CF1D9875E1B8102AD2C020A37

bQVUu4Bp9XrsudbAc2iYGl9cgSplbSD5mAsC3rsc/5XUi+Fe31nhXZKgIHfIui
2v.............................................................
f/NuPpeZ3KLUJGcpUGN4t393aaRXyoidSo4ekgUARJgnt/QND86zCyxJHyd7TmQS
-----END RSA PRIVATE KEY-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls ca cert
show cls summary

load tls device caIntermediates

Options for loading Transport Layer Security server Certifiate Authority Intermediates

Syntax

```
load tls device caIntermediates fromInput none|*

load tls device caIntermediates fromFile filename|none
```

Parameters

input

Type: STRING

String representing the Certficiate Authority Intermediates. The system will prompt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls device caIntermediates fromInput *
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE-----
MIIF1TCCA72gAwIBAgIBADANBgkqhkiG9w0BAQsFADB9MRswGQYJKoZIhvcNAQkB
..............RDz+0llBNWe2
-----END CERTIFICATE-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

show tls summary

load tls device cert

Options for loading Transport Layer Security server certification

Syntax

```
load tls device cert fromInput *

load tls device cert fromFile filename
```

Parameters

input

Type: STRING

String representing the server certification. The system will prompt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls device cert fromInput *
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE-----
MIIF1TCCA72gAwIBAgIBADANBgkqhkiG9w0BAQsFADB9MRswGQYJKoZIhvcNAQkB
..............RDz+0llBNWe2
-----END CERTIFICATE-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls server privateKey privKeyPass
show tls summary

load tls device privateKey

Options for loading Transport Layer Security server Private Key

Syntax

```
load tls device privateKey privKeyPass * fromInput *
```

```
load tls device privateKey privKeyPass * fromFile filename
```

Parameters

input

Type: STRING

String representing the Private Key. The system will promt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls device privateKey privKeyPass * fromInput *
Enter passphrase: ******
Enter PEM text (ctr-d to end):
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,16DD663CF1D9875E1B8102AD2C020A37

bQVUu4Bp9XrsudbAc2iYGl9cgSplbSD5mAsC3rsc/5XUi+Fe31nhXZKgIHfIui
2v.........................................................
f/NuPpeZ3KLUJGcpUGN4t393aaRXyoidSo4ekgUARJgnt/QND86zCyxJHyd7TmQS
-----END RSA PRIVATE KEY-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls server cert
show tls summary

load tls server caIntermediates

Options for loading Transport Layer Security server Certifiate Authority Intermediates

Syntax

```
load tls server caIntermediates fromInput none|*

load tls server caIntermediates fromFile filename|none
```

Parameters

input

Type: STRING

String representing the Certficiate Authority Intermediates. The system will prompt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls server caIntermediates fromInput *
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE-----
MIIF1TCCA72gAwIBAgIBADANBgkqhkiG9w0BAQsFADB9MRswGQYJKoZIhvcNAQkB
..............RDz+0llBNWe2
-----END CERTIFICATE-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

show tls summary

load tls server cert

Options for loading Transport Layer Security server certification

Syntax

```
load tls server cert fromInput *

load tls server cert fromFile filename
```

Parameters

input

Type: STRING

String representing the server certification. The system will prompt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls server cert fromInput *
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE-----
MIIF1TCCA72gAwIBAgIBADANBgkqhkiG9w0BAQsFADB9MRswGQYJKoZIhvcNAQkB
..............RDz+0llBNWe2
-----END CERTIFICATE-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls server privateKey privKeyPass
show tls summary

load tls server privateKey

Options for loading Transport Layer Security server Private Key

Syntax

```
load tls server privateKey privKeyPass * fromInput *

load tls server privateKey privKeyPass * fromFile filename
```

Parameters

input

Type: STRING

String representing the Private Key. The system will promt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
load tls server privateKey privKeyPass * fromInput *
Enter passphrase: ******
Enter PEM text (ctr-d to end):
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,16DD663CF1D9875E1B8102AD2C020A37

bQVUu4Bp9XrsudbAc2iYGl9cgSplbSD5mAsC3rsc/5XUi+Fe31nhXZKgIHfIui
2v.........................................................
f/NuPpeZ3KLUJGcpUGN4t393aaRXyoidSo4ekgUARJgnt/QND86zCyxJHyd7TmQS
-----END RSA PRIVATE KEY-----

Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

load tls server cert
show cls summary

logging

Used to set the level of detail captured by Trouble Reports and manually add text notes into log for Trouble report. To be used at direction of ZeeVee support team to aid in troubleshooting of issues.

Syntax

```
logging level arg
```

Parameters

arg

Type: INTEGER

Logging Level. Integer range from 1 to 4

Example

```
logging level 2
Success
```

Syntax

```
logging note string
```

Parameters

string

Type: text

String with length from 1 to 132 characters

Example

```
logging note "my inserted text"
Success
```

logout

Used to logout of the current session for force the logout of any other active session.

Syntax

```
logout force sessionId num
```

Parameters

num

Type: INTEGER

Session ID. Integer range from 1 to X, where X is the number is the session you wish to force a logout.

Examples

```
logout
Connection closed by foreign host.
```

```
logout force sessionId 2
Success
```

previewStream

Used to turn on/off a small thumbnail size preview stream that is viewable in the ZyPer Management Platform GUI. (ZyPer4K, ZyPerUHD60 and ZyPerUHD only) Note: Preview streams are not supported by the ZyPer4K-XS/XSE and ZyPer4K-XR

Syntax

`previewStream` *`enc arg comp`* `width` *`size`*

Parameters

`enc`

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

`arg`

argument	Description
`stop`	used to manually stop the preview stream. Note that it can turned back on from the GUI
`start`	used to manually start the preview stream.

`comp`

argument	Description
`hls`	set the format of the preview stream to HLS
`jpeg`	set the format of the preview stream images to JPEG

`size`

Type: Integer

Width of the preview stream in pixels. (180 to 400)

Example

```
previewStream MyEnc start hls width 300
BWG: After Start Preview Streams running = 3
Success


previewStream MyEnc stop
Success
```

HLS Notes

A maximum of 20 preview streams may be enabled at a single time.

ZyPer4K devices must be on firmware release 4.0.1.0 or newer for this feature to work.

The HLS stream can be viewed by any HLS capable viewer such as a browser. The path needed is shown below:

http://mp_ip_address/media/encoder_mac_address.m3u8

mp_ip_address is the IP address of the ZyPer Management Platform
encoder_mac_address is the MAC address of the Z4K encoder

Example

http://192.168.0.78/media/d8:80:39:eb:1c:ee.m3u8

JPEG Notes

JPEG images cannot be viewed in the ZyPer Management Platform GUI. This feature is intended for 3rd party control systems to grab individual JPEG images. (1 per second)

The JPEG images can be viewed by any JPEG capable viewer such as a browser. They can also be directly downloaded to a system. The path needed is shown below:

http://mp_ip_address/media/encoder_mac_address.jpeg

mp_ip_address is the IP address of the ZyPer Management Platform
encoder_mac_address is the MAC address of the Z4K encoder

Examples

http://192.168.0.78/media/d8:80:39:eb:1c:ee.jpeg

curl http://192.168.0.78/media/80:1f:12:4d:bb:11.jpeg > preview.jpg

redundancy add server ip

Adds new ZMP server for use in Redundancy configuration.

Syntax

```
redundancy add server ip ip
```

Parameters

ip

Type: STRING

IP Address of the server being added to the system for redundancy

Example

```
redundancy add server ip 172.16.6.125
Success
```

Related Commands

```
set server redundancy
redundancy switchover
redundancy delete downServers
redundancy delete server ip
```

redundancy delete downServers

Cleans up and removes any redundant servers from server list that are no longer available in the system.

Syntax

```
redundancy delete downServers
```

Parameters

none

Example

```
redundancy delete downServers
Success
```

Related Commands

set server redundancy
redundancy switchover

redundancy delete server ip

Deletes ZMP server for use in Redundancy configuration.

Syntax

```
redundancy delete server ip ip
```

Parameters

ip

Type: STRING

IP Address of the server being added to the system for redundancy

Example

```
redundancy delete server ip 172.16.6.125
Success
```

Related Commands

```
set server redundancy
redundancy switchover
redundancy delete downServers
redundancy add server ip
```

redundancy switchover

If there is an active slave, this command causes the existing master to become the slave and the existing slave to become the master. The server does not restart or re-initialize any other state, including any existing video and audio connections.

The IP address that is always assigned to the master. If the active slave becomes the master, this IP address will then terminate at that system. Note that any existing TCP connection will terminate and have to be reopened (to the new master).

Syntax

```
redundancy switchover
```

Parameters

none

Example

```
redundancy switchover
Success
```

Related Commands

```
set server redundancy
redundancy delete downServers
```

rename kvm

Changes the name of an existing kvm. (ZyPer4K-XS/XSE/XR family only)

Syntax

```
rename zone kvmname newName newkvmname
```

Parameters

kvmname

Type: STRING

The name of the existing kvm. String names are case-sensitive.

newkvmname

Type: STRING

The new name of the kvm. The name of the zone cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
Zyper$ rename kvm SouthWall newName NorthWall
Success
```

Related Commands

```
create kvm
delete kvm
show kvm
start kvm
stop kvm
```

rename zone

Changes the name of an existing zone

Syntax

```
rename zone zonename newName newzonename
```

Parameters

zonename

Type: STRING

The name of the existing zone. String names are case-sensitive.

newzonename

Type: STRING

The new name of the zone. The name of the zone cannot exceed 255 characters in length. Names are case-sensitive.

Example

```
Zyper$ rename zone FirstFloor newName SecondFloor
Success
```

Related Commands

create zone

delete zone

restart device

Restarts the specified device.

Syntax

```
restart device id
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Examples

```
restart device myEncoder2
Success

restart device 0:1e:c0:f6:cb:76
Success
```

Related Commands

`shutdown server reboot`

restore server database

Restores a stored server database from file. (Stored on the ZyPerMP hardware)

Important Note: Saved database to be restored MUST have been created using the exact same version of API that is currently running.

Syntax

`restore server database name`

Parameters

name

Type: STRING

The name of the stored database. Names are case-sensitive.

Example

```
restore server database jan16_2019
Loaded database jan16_2019; restarting server
Success
```

Related Commands

`save server database`

revert server

Returns to a previously installed version of the API and device database.

This feature can be used to go back to a previous software version and database version in case of a failed software upgrade. Primarily used to recover previous state if something goes wrong.

Syntax

```
revert server
```

Note: The show sever info command will identify the Previous Version that will be restored to the system.

Example

```
revert server
Reverting from update_nuc_1.8.34605.zyper to update_nuc_2.0.34928.
zyper
Success
```

Related Commands

`show server info`

run preset

Manually executes an existing preset

Syntax

```
run preset name
```

Parameters

name

Type: STRING

The name of the existing preset. Names are case-sensitive.

Example

```
run preset lunch
Success

run preset closing
Success
```

Related Commands

create preset
delete preset
set preset
show preset

save deviceEdid

Saves the EDID of the downstream sink to the `srv/ftp/files` folder on the Management Server. Executing this command will generate two file types: `.bin` and `.txt`. The `.bin` file is the EDID is standard format. The `.txt` file is the decoded EDID data.
See Using Custom EDID Data (page 96) for more information on using this command.

Syntax

```
save deviceEdid id file
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder that is connected to the sink device. String names are case-sensitive.

file

Type: STRING

The name of the EDID file. Two files will be created using the *file* name: `.txt` and a file with no extension.

Example

```
save deviceEdid 0:1e:c0:f6:a5:2f myEDID
Success
```

Related Commands

`load encoderEdid`
`set server autoEdidMode`

save server database

Saves the current MP database to a file. (Stored on the ZyPerMP hardware)

Syntax

```
save server database name
```

Parameters

name

Type: STRING

The name of the database. Names are case-sensitive.

Example

```
save server database jan16_2019
Saved database to jan16_2019
Success
```

Related Commands

`restore server database`

save system config

Saves the current system configuration to a file. (Stored on the ZyPerMP hardware)

Syntax

```
save system config name
```

Parameters

name

Type: STRING

The name of the file. Names are case-sensitive.

Example

```
save system config march24
Saved config to /srv/ftp/files/march24
Success
```

Related Commands

save server database
restore server database

script

Executes the specified script. The script must exist in the `/srv/ftp/files` folder. Use the optional `loop` argument to place the script in a loop. The script will continue running until a key is pressed on the keyboard.

Syntax

`script` *`file`* `[loop]`

Parameters

`file`

Type: STRING

The name of the script file.

Example

```
script myScript
Success
```

Related Commands

`sleep`

send

Sends an IR, RS232 or CEC string to the specified device. Use the *type* parameter to specify an IR, RS232 or CEC code.

Syntax

```
send id type text
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device.

type

Type: STRING

Specifies IR, CEC or RS232 command

argument	Description
`ir`	The string must be the hex representation of the binary data. (Pronto code) The maximum length for a string is 1024 characters. (ZyPer4K family only)
`cec`	on | off (Used to turn a device on or off)
`cec hexString`	hex-numerals-no-delimiters (ZyPer4K and ZyPerUHD60 families only)
`rs232`	The string is ASCII and must not exceed 256 characters in length. Spaces and the following control characters are supported as a portion of the string: `\n` New line `\r` Carriage return `\t` Tab `\\` Slash `\xnn` Hex value, where `nn` is a two-digit hex value, including leading zeros

text

Type: STRING

The string to send. See the table, above, for restrictions.

Example

```
send myDecoder2 ir 0000006900000015005f0017003000170030001700300017003000170030001
700300017001700170030001700170017001700170030001700170017003000170
030001700170017003000170017001700170017001700170030001700300017003
00200
Success

send myDecoder2 rs232 ZeeVee_support_is_the_greatest\r\n
Success

send myDecoder2 cec on
Success

send myDecoder2 cec off
Success
```

Important Notes

CEC is not supported on ZyPerHD

CEC functionality on the ZyPer4K is only supported with hardware firmware version 3.5.2 and newer.

CEC hexString command is not supported on ZyPerUHD

Related Commands

`set device rs232`

set account all

Sets various security features for all accounts

Syntax

```
set account all option
```

Parameters

option

Type: STRING

The security feature to configure

argument	Description
`authMode`	Sets telnet or web authorization. (telnet oldAuth|fullAuth), (web backend|browser)
`concurrentSessionsMax`	Maximum number of sessions allowed. <int>|unlimited
`idleLogout minutes`	Number of idle minutes before a logout is forced. <int>|unlimited (Note: authMode web must be set to "backend" to use this feature.)
`onThreeFailures`	What to do if login attempt fails 3 consecutive times. lockoutMinutes <int>|none disableAccount true|false
`password`	Set complexity or duration of passwords. complex enabled|disabled minLen <int> duration initialExpire enabled|disabled minDays <int> maxDays <int>|unlimited

Examples

```
set account all authMode telnet oldAuth
Success
set account all authMode web backend
Success
set account all concurrentSessionsMax 5
Success
set account all idleLogout minutes unlimited
Success
set account all onThreeFailures lockoutMinutes none disableAccount
false
Success
set account all onThreeFailures lockoutMinutes 1 disableAccount
true
Warning:(6) You set both actions for onThreeFails. Only setting
disableAccount true
Success
set account all password complex disabled minLen 8
Success
```

set account password

Used to change existing accounts password

Syntax

```
set account password existing currentpass |* newpass
```

Parameters

option

Type: STRING

The account feature to configure

argument	Description
`currentpass`	Current password (Case sensitive) Can also use wildcard * to be prompted for password
`newpass`	New password (Case sensitive)

Examples

```
set account password existing redsox new yankees
Success

set account password existing * new redsox
Existing password: *******
Success
```

set account username

Sets various features associated to a specific account

Syntax

```
set account username option
```

Parameters

option

Type: STRING

The account feature to configure

argument	Description
`2fa`	Enable or disable 2 factor authorization (enabled|disabled)
`expirePassword`	Set password to expire or not (enabled|disabled)
`lock`	Lock a specific account
`password new`	Set a new password for an account (<string>|*)
`role`	Assign an existing role to this account (<rolename>)
`unlock`	Unlock a specific account

Examples

```
set account username ArtW 2fa enabled
2fa-secret: GOCE5AMI6ZP7NNVZTWUK2375UQ
Success
set account username ArtW 2fa disabled
Success
set account username ArtW expirePassword enabled
Success
set account username ArtW lock
Success
set account username ArtW unlock
Success
set account username ArtW role admin
Success
```

set decoder

Sets the audio output type and video timing details for the specified decoder.

Syntax

```
set decoder id mode type
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

mode

Type: STRING

argument	Description
`analogAudioOut`	Audio output from the Audio port on the decoder. (ZyPer4K and ZyPerUHD60 family only)
`connectionMode`	Sets/changes current connection mode to decoder. (Options are fast-switched, genlocked and genlocked-scaled) (ZyPer4K family only)
`displayAdvancedTiming`	Set advanced features, Front porch, sync width, sync polarity and total size
`displayMode`	Set display to box, crop or stretch input stream within display resolution
`displayResolution`	Set display resolution manually (pixels) or automatically based on EDID.
`hdcpMode`	Allows user to force HDCP protection at level 1.4 or 2.2 on previously unprotected content. (ZyPerUHD and ZyPerUHD60 only) Will cause decoder to reboot.
`hdmiAudioOut`	Audio output from the HDMI port on the decoder. (ZyPer4K and ZyPerUHD60 family only)

type

Type: STRING

HDCP options. (Note: Valid with ZyPerUHD and ZyPerUHD60 only) Used to minimize connection time.

argument	Description
`auto`	Maintain existing HDCP level. None if none
`forceVersion1.4`	Apply HDCP 1.4 protection to output stream
`forceVersion2.2`	Apply HDCP 2.2 protection to output stream

The audio mode (analog out or HDMI out).

argument	Description
`source analogAudio`	Uses the audio output created with the join command. (ZyPer4K only)
`source hdmiAudio`	Uses the HDMI stream (HDMI audio-out only) Use if video in Fast-Switch mode. (ZyPer4K only)
`source hdmiPassthroughAudio`	Used if video is in Genlock mode. (ZyPer4K only)
`source hdmiAudioDownmix`	Uses the HDMI-downmix stream
`danteAudio`	Use stream that Decoder is receiving from encoder that was in Dante format. (ZyPer4K-XSE only) (Outside Dante Source routed to Encoder then routed to Decoder)
`directDanteAudio`	Use Dante Audio Stream routed to Decoder from outside Dante source.

Display timing, aspect ratio, mode, size.

argument	Description
`syncFrontPorch`	Synchronization mode.
`syncWidth`	Synchronization width
`hsyncPolarity`	Horizontal sync polarity (auto, negative, positive)
`vsyncPolarity`	Vertical sync polarity (auto, negative, positive)
`totalSize`	Horizontal and vertical size (Pixels or auto)
`box`	Box image within display. (Smaller source to larger display)
`crop`	Crop image within display (Larger source to smaller display)
`stretch`	Scale image to fill display. (Scale up or down) (Default Setting)
`pixelsHoriz`	Width in pixels or auto
`pixelsVert`	Height in pixels or auto
`fps`	Frames per second
`source`	Match decoder resolution to source input size
`auto`	automatically based on EDID

Command Description: Override output display size and fps

set decoder <Decoder_Name or MAC> displayResolution activeSize <int> pixelsHoriz <int> pixelsVert <int> fps <int>|source

This command allows an override of EDID parameters supplied by the display. Regardless of what the supplied EDID indicates, the decoder will generate a stream with specified overall size and frame rate parameters.

Note that in "genlock-scaled" mode, the frame rate parameter is ignored – it must be the same as the encoder frame rate. This does mean care must be taken when setting this parameter if the source stream is 60fps (e.g. 720p60fps) and scaled to 4K. That only works if the display supports 4K60.

If configured resolution specification in these parameters that exceed the displayed maximum resolution, the display will black out with no indication to the user.

Example command:
Zyper$ set decoder Dec1 displayResolution activeSize 3840 2160 fps 60

Command Description: Output display size determined by received EDID

Command Syntax
set decoder <Decoder_Name or MAC> displayResolution auto

The command causes the decoder to set output display size to the "preferred" value in the EDID received from the display.

Command Description: Override detailed video parameters

Command Syntax
set decoder <decoderMac|decoderName> displayAdvancedTiming activeSize <pixelsHoriz:int> <pixelsVert:int> fps <float> total-size <pixelsHoriz:int> <pixelsVert:int> syncFrontPorch <pixelsHoriz:int> <pixelsVert:int> syncWidth <pixelsHoriz:int> <pixelsVert:int> syncPolarity hPositive|hNegative vPositive|vNegative

This command allows an override of EDID parameters supplied by the display. Regardless of what the supplied EDID indicates, the decoder will generate a stream with specified detailed timing parameters.
If configured resolution specification in these parameters that exceed the displayed maximum resolution, the display will black out with no indication to the user.

Example command:
Zyper$ set decoder Dec1 displayAdvancedTiming activeSize 1920 1080 fps 60 totalSize 2200 1200 syncFrontPorch 88 4 syncWidth 44 5 syncPolarity hPositive vPositive

set decoder autoAudioConnections hdmiAudioFollowVideo

Tells the decoder to automatically join Audio associated with connected Video stream or not.

Syntax

```
set decoder id autoAudioConnections hdmiAudioFollowVideo arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Audio will follow video automatically
`disabled`	Audio will not follow video automatically

Example

```
set decoder myDecoder autoAudioConnections hdmiAudioFollowVideo
enabled
Success
```

set decoder danteAudioOut

Sets the ZyPerUHD60 Dante enabled decoder to either Transmit or Receive mode. (Note that unit will reboot after making a change). Also will select what received audio stream to convert to Dante when in Transmit mode.

Syntax

```
set decoder id danteAudioOut source arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`joinedAudio`	Makes the UHD60 a Dante Transmitter (ZyPerUHD60 only)
`none`	Makes the UHD60 a Dante Receiver (ZyPerUHD60 only)
`analogAudio`	Convert analog audio to Dante format (ZyPer4K-XS only)
`hdmiAudioDownmix`	Convert HDMI downmix audio to Dante format (ZyPer4K-XS only)
`DanteAudio`	Use the Dante audio stream that is received from Encoder (ZyPer4K-XS only)

Examples

```
set decoder MyDecoder1 danteAudioOut source none
Warning:(36) Device has been restarted
Success

set decoder myDecoder danteAudioOut source joinedAudio
Warning:(36) Device has been restarted
Success
```

set decoder edidPreferMode

Sets the preferred resolution from the display EDID

Syntax

```
set decoder id mode type
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

mode

Type: STRING

argument	Description
`edidPreferMode`	Select preferred EDID mode

type

Type: STRING

HDCP options. (Note: Valid with ZyPerUHD only) Used to minimize connection time.

argument	Description
`max`	Default mode. Selects the largest resolution defined in the EDID.
`strict`	Selects the Preferred resolution as stated in the display EDID

"max" – Default mode. Selects the largest resolution defined in the EDID. This has been the operating mode prior to this command. In almost all cases, this is the native resolution of the display. However, some displays can accept a resolution above the native (and scale down). In this case, it is better to use the "strict" mode.

"strict" – The Preferred Resolution is selected as defined in the EDID 1.3 specification. EDID 1.3 specifies that the first Detailed Timing Descriptor in the Standard Timing Information block is always the preferred resolution, although it is only the native resolution if the native-resolution flag is set. If the native-resolution flag is not set, then the maximum resolution will be chosen (falls back to "max" mode).

Note: All comparisons of "resolution" actually mean comparisons of the associated Pixel Clock. The Pixel Clock represents the entire resolution definition: horizontal and vertical size, fps, bit-depth and color decimation (RGB/4:4:4, 4:2:2, 4:2:0).

The command will immediately reanalyze the active EDID and if needed change the preferred resolution and reconnect to the encoder.

The reason for the "max" mode, and for it being the default, is that many displays do not follow the EDID 1.3 specification, claiming a native, Preferred Resolution below the display's actual native resolution. It is fairly common for a UHD display to have an HD resolution as the specified preferred resolution.

Note: ZyPer4K and ZyPerUHD, depending on mode, may support only a limited set of output resolutions, particularly when the scaler is enabled. ZMP will choose the active resolution based decoder capability, scaler mode and preferred resolution. However, the display's Preferred Resolution is displayed regardless of what the decoder ultimate actually uses. The active resolution is displayed in the decoder status as well.

Overriding Preferred Resolution Selection

It should rarely be required. But if the EDID supplied by the display is not correct, or for some reason ZMP chooses a Preferred Resolution that is not desired, the following command will force the decoder to a specific output resolution:

```
        set decoder <decoder> displayResolution activeSize <int>
<int> fps <float>
```

When set, the decoder output resolution will remain as specified without exception.

Note: When in this mode, it is very possible that no video will be displayed, and with no warning from ZMP. It is up to the user to ensure that the output settings are valid for the display.

Scaler Control

ZyPer4K, ZyPerUHD60 and ZyPerUHD decoders have output scaling. Besides the obvious benefit of supporting HD-only displays with a UHD source, the other major benefit is faster switching times. With ZyPer4K, there is virtually no delay. With ZyPerUHD it is less than a one second.

However, there are some cases where disabling the scaler produces a better image. Of course, if the scaler is disabled and the source provides a resolution greater than the display's ability, it will be black. To solve this problem, we have a new mode that disables the scaler, but only if the display can handle the source resolution.

The decoder display-resolution command now has an option called "source".

```
set decoder <decoder> display-resolution source
```

When in "source" mode the scaler is disabled if the display can handle the received resolution. Otherwise it is automatically enabled (e.g. if the source is 480 and the Preferred Resolution is 1080 then the scaler is disabled, but if source is UHD and the Preferred Resolution is 1080, then the scaler is enabled).

The downside to this mode: switching time between non-scaled resolutions is about 3 seconds. Switching time between scaled and non-scaled resolutions is closer to 4s.

Active Output Resolution Selection

Selecting the correct output resolution for a decoder is, unfortunately, a fairly complicated endeavor. Clearly depends on the display (Preferred Resolution), but also on the decoder capability and the source resolution.

Remember: All comparisons of "resolution" actually mean comparisons of the associated Pixel Clock. The Pixel Clock represents the entire resolution definition: horizontal and vertical size, fps, bit-depth and color decimation (RGB/4:4:4, 4:2:2, 4:2:0).

Also, setting "edidPreferMode" only affects which Preferred Resolution is chosen. It does not affect when that Preferred Resolution is used (or if it is used). Although the chosen Preferred Resolution is always reported in the decoder status output (as is the chosen active output resolution).

ZyPer4K HDMI 2.0 Devices

Presently, the decoder active resolution is limited to a number of resolutions: 4096x2160, 3840x2160, 1080x1920 or 1280x720. The closest lower resolution is used.

There are a number of exceptions to the operation.
• Scaler always converts to 8bit 444/RGB. That means UHDp60 4:2:0 is converted to UHDp60 4:4:4. UHDp60 YUV 4:2:0 bit rate is lower than HDM 1.4. But UHDp60 4:4:4 is not. In this case, the output FPS is divided by 2.
• If in genlockScaled, videoWall or window mode, decoder FPS must equal encoder FPS
o Means 1080p60 scaling to UHD must be UHDp60, which won't work if display is only UHDp30 capable.
o If UHDp60 > decoder Preferred Resolution, then the output is left at 1080p60.
• If source is 1080i
o Output must be input FPS * 2
o If decoder resolution > 1080, it is set to 1080.

displayResolution = auto

When in this mode, the output resolution will always be the Preferred Resolution. There really is no reason not to use this mode with the Z4K Charlie and will produce the lowest switching times.

displayResolution = source

When in this mode, the output resolution will always be the encoder resolution, unless the source resolution greater than the encoder resolution (same case as displayResolution auto).

This mode may provide better video at or below the preferred resolution of the display. However, the switching time is somewhat slower (~3.3s).

displayResolution = sourceIgnoreEdid

Same operation as displayResolution = auto, but effectively using a manually entered Preferred Resolution. Generally only used if the EDID is incorrect.

ZyPerUHD

The ZyPerUHD scaler scales up fine (source resolution lower than display preferred). However, it can only scale down from UHD to 1080.

Even with this limitation, the vast majority of installations will be fine. The exception comes with PC-based resolutions. For example a case that will not work well:
• 1080-only display and source resolution of 1920x1200

For the cases where VESA/PC resolutions such as 1920x1200, 2560x1440 and 2560x1600 are needed, all displays must be at least that resolution or greater. For example, a 1920x1200 display can handle all resolutions up to 1920x1200 and it can also handle UHD, since the decoder will output UHD scaled down to 1080 (which is fine for a 1920x1200 display).

And, clearly, all of those resolutions will be fine if the displays are UHD capable (scaling up works, plus, the new mode "display-size source" can be used).

If a configuration that causes downscaling that is not handled well, likely generating poor video, a warning will be generated.

displayResolution = auto

When in this mode, the output resolution will always be the Preferred Resolution, unless the source resolution greater than the preferred resolution.

This mode provides the fastest switching time (less than 1 second). However, there may be some cases where video quality is less than when using display-resolution = source.

If source is greater than decoder Preferred Resolution, then decoder output will be 1920x1080 (unless the display does not support it) with the preferred FPS. As noted, the only case this normally works for is when the source is 3840x2160.

displayResolution = source

When in this mode, the output resolution will always be the encoder resolution, unless the source resolution greater than the encoder resolution (same case as displaySize auto).

This mode may provide better video at or below the preferred resolution of the display. However, the switching time is somewhat slower (~3.3s).

displayResolution = sourceIgnoreEdid

Same operation as displayResolution = auto, but effectively using a manually entered Preferred Resolution. Generally only used if the EDID is incorrect.

set decoder hdmi5vControl

Enables or disables 5V HDMI line of the decoder. (ZyPer4K-XS, ZyPer4K-XSE and ZyPer4K-XR only)

When decoder is not receiving a video stream the decoder will disable the 5V HDMI line.

Syntax

```
set decoder id hdmi5vControl arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	5V HDMI line will disable when no video streamed ot the decoder.
`disabled`	5V HDMI line is never disabled. (Default)

Example

```
set decoder myDecoder hdmi5vControl enabled
Success
```

If you attempt to run this command on a decoder that is not XS/XSE/XR or not on correct firmware you get the following error.

```
Error:(29) Device myDecoder does not support or cannot change:
videoPort with value hdmi5vControl.
```

Notes

ZyPer4K-XS or ZyPer4K-XR must be updated to firmware version 1.3.2.4 or newer for this command to work.

The connection before disconnecting video from the decoder must be "genlocked" to fully disable video and cut the 5V line.

set decoder lowLatency

Enables or disables low latency mode for the ZyPerUHD60 decoder. (ZyPerUHD60 only)

Syntax

```
set decoder id lowLatency arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	enable low latency mode. (Default)
`disabled`	disable low latency mode.

Example

```
set decoder myDecoder lowLatency enabled
Success
```

Notes

There may be instances where the user will want to disable low latency mode. Example: We have found a compatiblity issue with a Crestron HD-DA4-4KZ-E splitter that requires low latency mode disabled in order to properly recognise the input from ZyPerUHD60 decoder.

set decoder osdStatusMode

Enables or disables on-screen-display feature of the decoder. (ZyPerUHD and ZyPerUHD60 only)

When decoder is not receiving a stream the decoder will display a "No Source Found" screen. In the lower corner of this screen is displayed the following information:

Firmware version and date
IP address of the decoder
Remote IP: (Encoder it is attempting to get stream from if any)
MAC Address

The osdStatusMode command will make this information visible or not.
Note, changing status with the command will force the decoder to reboot.

Syntax

```
set decoder id osdStatusMode arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	OSD feature enabled.
`disabled`	OSD feature disabled.

Example

```
set decoder myDecoder osdStatusMode enabled
Warning:(36) Device myDecoder has been restarted
Success
```

set decoder powerSave

Enables or disables power save feature of the decoder. (ZyPerUHD and ZyPerUHD60 only)

When decoder is not receiving a stream the decoder will enter a low power mode and the display will go black.

Syntax

```
set decoder id powerSave arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Power save feature enabled.
`disabled`	Power save feature disabled.

Example

```
set decoder myDecoder powerSave enabled
Success
```

set device dante port

Sets port for Dante activity. (ZyPerUHD60-2EA/2DA and ZyPer4K-XSE units only)

Used to set the Dante port to either Video Port or Utility Port.

Syntax

```
set device id dante port arg boot
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`video`	Sets Dante to the Video Port and LAN
`utility`	Sets Dante to the Utility Port and LAN

boot

Type: STRING

Supply one of the following arguments.

argument	Description
`reboot`	Reboot the endpoint
`noReboot`	No not reboot the endpoint

Example

```
set device UHD60-2EA dante port utility reboot dhcp
Success
```

Related Commands

```
set device dante ip
set device dante ip static
```

set device dante ip

Sets DHCP mode for the specified device. (ZyPerUHD60-2EA and 2DA only)
Note only applies when ZyPerUHD60-2EA/2DA using using Dante on the Utility Port.
(Separate LAN from primary AV network)

In future this command will also support Dante enabled ZyPer4K-XS units.

Syntax

```
set device id dante ip arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`dhcp`	IP address assigned by DHCP server
`linkLocal`	IP address self assigned Link-Local

Example

```
set device UHD60-2EA dante ip dhcp
Success
```

Related Commands

```
set device general name
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device dante ip static

Sets static mode for the specified device. The IP address, subnet mask, and gateway must be supplied. (ZyPerUHD60-2EA and 2DA only)

Note only applies when ZyPerUHD60-2EA using using Dante on the Utility Port. (Separate LAN from primary AV network)

In future this command will also support Dante enabled ZyPer4K-XS units.

Syntax

```
set device id  ip static addr mask gateway
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

addr

Type: IP Address

The desired IP address for the device.

mask

Type: IP Address

The desired subnet mask for the device.

gateway

Type: IP Address

The desired gateway for the device.

Example

```
set device UHD60-2EA dante ip static 10.5.68.121 255.255.255.0
10.5.64.1
Success
```

set device dante vlan mode

Sets Dante VLAN. (KDS-17 units only)

Used to enable/disable Dante VLAN and set VLAN ID if enabled.

Syntax

```
set device id dante vlan mode arg vlanId LANid boot
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Enable Dante VLAN mode
`disabled`	Disable Dante VLAN mode

LANid

Type: Integer from 1 to 4000

VLAN ID for Dante.

boot

Type: STRING

Supply one of the following arguments.

argument	Description
`reboot`	Reboot the endpoint
`noReboot`	No not reboot the endpoint

Example

```
set device KDS17-Dec1 dante vlan mode enabled vlanId 3 reboot
Warning:(36) Device has been restarted
Success
```

Related Commands

```
set device dante ip
set device dante ip static
```

set device general name

Sets the name for the specified encoder or decoder.

Syntax

```
set device id  general name str
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

Important Note: The following characters are not valid for device names.

Colon :
Quotes "
Blank Spaces

str

Type: STRING

The name for the device.

Example

```
set device myDecoder5 general name Samsung-55
Success
```

Related Commands

```
set device ip
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device ip

Sets DHCP mode for the specified device.

Syntax

```
set device id ip arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`dhcp`	IP address assigned by DHCP server
`linkLocal`	IP address self assigned Link-Local

Example

```
set device ABC ip dhcp
Success
```

Related Commands

```
set device general name
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device ip static

Sets static mode for the specified device. The IP address, subnet mask, and gateway must be supplied.

Syntax

```
set device id  ip static addr mask gateway
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

addr

Type: IP Address

The desired IP address for the device.

mask

Type: IP Address

The desired subnet mask for the device.

gateway

Type: IP Address

The desired gateway for the device.

Example

```
set device ABC ip static 10.5.68.121 255.255.255.0 10.5.64.1
Success
```

Related Commands

```
set device general name
set device ip
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device irProcessing

Configures ZyPer4K endpoint to process input IR commands to issue channel up or channel down API command.

ZyPer Remote is and IR remote control. Part number: ZVREMOTE
Hitting Up or CH+ button will issue channel up API command.
Hitting Down or CH- button will issue channel down API command.

ZeeVee IR Receiver is required to be plugged into Decoder IR input port.
Part number: Z4KIRRX

ZyPer Trigger is a device to connect a "button" to the ZeeVee decoder IR ports.
Part number: Z4KIRTRIGTX

Syntax

```
set device id irProcessing arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`zyperTrigger`	Process button press from ZyPer Trigger device
`zyperRemote`	Process up/down button press from ZeeVee IR remote control
`none`	Do not process IR inputs

Example

```
set device Z4KDec irProcessing zyperRemote
Success
```

Related Commands

```
set device general name
set device ip static
```

set device optionCard

Manually forces option card type for ZyPer4K extended box Encoders (ZyPer4K only).

Typically extended box encoders will automatically identify themselves to the ZMP. However in some cases using the "auto" setting, the expansion/option card is not detected. Manually setting the feature forces correct operation. Note you must select the correct setting based on the ZyPer4K encoder hardware you have. For example 12G SDI hardware cannot be set to Display Port.

Syntax

```
set device id optionCard type arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`auto`	ZMP will automatically detect the proper setting. Note issue above.(default)
`hdsdi`	Force setting to HD-SDI
`displayPort`	Force setting to DisplayPort
`analog`	Force setting to Analog
`hdmiOptionalIn`	Force setting to Dual HDMI input with loopout
`sdi12g`	Force setting to 12G SDI input

Example

```
set device ABC optionCard type sdi12g
Success
```

Related Commands

`set device general name`
`set device ip static`

set device rs232

Sets the RS232 settings for the specified device.

Syntax

```
set device id rs232 baud data stop parity
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

baud

Type: INTEGER

The baud rate for the device. Supply one of the following values from the table below.

argument		
2400	9600	19200
38400	57600	115200

data

Type: INTEGER

The data bit setting for the device. Supply one of the following values from the table below.

argument
7-bits
8-bits

stop

Type: INTEGER

The stop bit setting for the device. Supply one of the following values from the table below.

argument
`1-stop`
`2-stop`

parity

Type: STRING

The parity setting for the device. Supply one of the following values from the table below.

argument
`even`
`odd`
`none`

Example

```
set device decoderNumber2 rs232 57600 8-bits 1-stop none
Success
```

Related Commands

```
send
set device general name
set device ip
set device ip static
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device security

Mechanism to enable security over Semtech's server-device communication. First, there has to be an overall key associated with the server (deviceSecurityKey). Then, each device has to enable the security. Provides authentication and encryption. This only works with ZyPer4K-XS/XSE and ZyPer4K-XR devices. Once a device has been enabled for a specific server, it will not work with any server without the same key. Redundancy automatically sets the same key on both servers. If the key is lost, devices have to be hardware factory defaulted

Syntax

```
set device id security arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Feature enabled.
`disabled`	Feature disabled.

Example

```
set device Encoder_1 security enabled
Success
```

Related Commands

`set server security deviceSecurityKey`

set device sendIpMcastRange

Sets allowable range of multicast addresses for selected devices. (ZyPer4K family only)

Syntax

```
set device id sendIpMcastRange first:ip last:ip
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive. Can all use "all" or "encoders" as an ID option.

first:ip / last:ip

Type: Multicast Address

Supply the starting and ending multicast addresses in the allowable range.

Note: Allowable range is from 224.1.1.1 to 239.255.255.255

Example

```
set device encoders sendIpMcastRange 224.1.1.25 224.1.2.125
```

Related Commands

```
set device general name
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device sourceDisplay iconImageName

Assigns an icon to the desired device. The icon will be displayed within the ZMP to identify the device.

Syntax

`set device` *`id`* `sourceDisplay iconImageName` *`fname`*

Parameters

`id`

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder. String names are case-sensitive.

`fname`

Type: FILENAME

The full filename of the icon to be used. The filename is case-sensitive.

argument	Description
`abc`	ABC network icon
`cbs`	CBS network icon
`nbc`	NBC network icon
`fox`	Fox network icon
`xbox`	Xbox game console icon
`golf`	Golf channel icon
`espn`	ESPN network icon
`tennis`	Tennis channel icon
`cnn`	CNN network icon
`ps3`	PlayStation game console icon
`DVD`	DVD player icon
`BluRay`	BluRay icon
`VCR`	VCR icon
`CableBox`	Cable box icon
`Laptop`	Laptop icon
`BroadcastCamera`	Broadcast camera icon
`SecurityCamera`	Security camera icon

Example

```
set device Encoder1 sourceDisplay iconImageName cbs
Success
```

Related Commands

```
set device general name
set device ip
set device ip static
set device rs232
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device sourceDisplay location

Assigns a location description for the specified device.

Syntax

```
set device id sourceDisplay location loc
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

loc

Type: STRING

The location description of the device (e.g. "Conference_Rm", "Den", etc.).
Do not use quotes when specifying this string value.

Example

```
set device myDecoder3 sourceDisplay location VideoWall-1
Success
```

Related Commands

```
set device general name
set device ip
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device sourceDisplay manufacturer

Assigns a manufacturer description for the specified device.

Syntax

```
set device id sourceDisplay manufacturer mfg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

mfg

Type: STRING

The manufacturer description of the device (e.g. "Sony", "Panasonic", etc.).
Do not use quotes when specifying this string value.

Example

```
set device myDecoder3 sourceDisplay manufacturer Sony
Success
```

Related Commands

```
set device general name
set device ip
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device sourceDisplay model

Assigns a model description for the specified device.

Syntax

```
set device id sourceDisplay model model
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

model

Type: STRING

The manufacturer's model number of the device.
Do not use quotes when specifying this string value.

Example

```
set device myDecoder3 sourceDisplay model DVPSR210P
Success
```

Related Commands

```
set device general name
set device ip
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay serialNumber
```

set device sourceDisplay serialNumber

Assigns the manufacturer serial number for the specified device.

Syntax

```
set device id sourceDisplay serialNumber serial
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

serial

Type: STRING

The manufacturer serial number of the device.

Example

```
set device myDecoder3 sourceDisplay serialNumber 123456789
Success
```

Related Commands

```
set device general name
set device ip
set device ip static
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
```

set device usbFilter

Allows restrictions to USB use on selected device. (ZyPer4K with Icron only. Not supported on ZyPer4K-XS or ZyPer4K-XR units)

Syntax

```
set device id usbFilter arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder. String names are case sensitive

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`none`	No restrictions on USB port
`exceptHid`	Only HID devices will work
`storage`	Allows any USB device except Storage devices

Example

```
set device myDecoder2 usbFilter none
Success
```

Related Commands

```
set device general name
set device ip
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device usbType

Used to select the USB function on ZyPer4K-XSE units. (ZyPer4K-XSE units with EAU or DAU suffix only)

ZyPer4K-XSE units with EAU or DAU suffix have the ability to operate with either the built in HID USB feature from Semtech or the full USB 2.0 feature from Icron. Both cannot function at the same time. This command toggles between the two options. This can be important to ensure compatibility between other ZyPer4K family endpoints. For example the White Box ZyPer4k units with USB only have the Icron option. The ZyPer4K-XS units and ZyPer4K-XSE units without EAU/DAU suffix only have the Semtech option.

Syntax

```
set device id usbType arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder or decoder. String names are case sensitive

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`full`	Use the Full USB 2.0 option from Icron
`hid`	Use the HID only option from Semtech

Example

```
set device myDecoder2 usbType full
Success
```

Note: Can only change between full and hid mode if no active USB connection between encoder and decoder

```
set device XSE_Enc usbType full
Error:(226) Existing USB connection; cannot change usb type
```

Related Commands

`set device usbFilter`

USB Interoperability Chart

			Decoders						
			Z4K-XS	Z4K-XSE	Z4K-XSE Dante+Icron	Z4KXS-WallPlate	Z4KXS-WallPlate Icron	Z4K-Icron	Z4K-XR
	USB Routing	**Applicable Models**	Z4KDECC3XS Z4KDECF3XS	Z4KCXS-D Z4KFXS-D Z4KC2XS-D Z4KF2XS-D	Z4KCXS-DAU Z4KFXS-DAU Z4KC2XS-DAU Z4KF2XS-DAU	Z4KDECC3XS-WPUS	Z4KDECC3XS-WPUS-I	Z4KDECF3U Z4KDPDECF3U Z4KDECC3U Z4KDPDECC3U	Z4KDECC3XR Z4KDECFSXR Z4KDECFLXR
Encoders	**Z4K-XS**	Z4KENCC3XS Z4KENCF3XS	Yes	Yes	Yes (Disable Icron)	Yes	No	No	Yes
	Z4K-XSE	Z4KC2XS-E Z4KF2XS-E	Yes	Yes	Yes (Disable Icron)	Yes	No	No	Yes
	Z4K-XSE Dante+Icron	Z4KC2XS-EAU Z4KF2XS-EAU	Yes (Disable Icron)	Yes (Disable Icron)	Yes	Yes (Disable Icron)	Yes	Yes	Yes (Disable Icron)
	Z4KXS-WallPlate	Z4KENCC3XS-WPUS	Yes	Yes	Yes (Disable Icron)	Yes	No	No	Yes
	Z4KXS-WallPlate-Icron	Z4KENCC3XS-WPUS-I	No	No	Yes (USB 2.0 only)	No	Yes	Yes	No
	Z4K-Icron	Z4KENCF3U Z4KENCC3U Z4K12GSDIENCF3U Z4K12GSDIENCC3U Z4KDPENCF3U Z4KDPENCC3U Z4KANLENCF3U Z4KANLENCC3U Z4KHDMIENCC3U Z4KHDMIENCF3U	No	No	Yes (USB 2.0 only)	No	Yes	Yes	No
	Z4K-XR	Z4KENCC3XR Z4KENCF3XR Z4KENCFLXR	Yes	Yes	Yes (Disable Icron)	Yes	No	No	Yes

USB not supported between devices
USB HID supported between devices (may need Icron to be disabled on XSE)
Full USB 2.0 supported between devices
USB Host port does not support direct connection to a USB 3.0 port. USB Hub is required between encoder and PC

set device utilityPort

Enables or disables the 1Gb Utility Ethernet port on the specified encoder or decoder. (ZyPer4K and ZyPerUHD60 only)

Syntax

```
set device id utilityPort arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Ethernet port is enabled.
`disabled`	Ethernet port is disabled (ZyPer4K only)
`onlyDanteAudio`	Ethernet port is enabled as a sepate LAN with Dante TX/RX. (ZyPerUHD60-2EA and ZyPerUHD-2DA only)

Example

```
set device myDecoder5 utilityPort disabled
Success
```

Related Commands

```
set device general name
set device ip
set device rs232
set device sourceDisplay iconImageName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

set device videoPort

Selects active input port for ZyPer4K units with multiple inputs. (ZyPer4K and ZyPerUHD60 only)

Syntax

```
set device id videoPort arg
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`hdmi`	Use the HDMI input (Z4K Located to the right) "HDMI IN1" on UHD60
`hdmiOptionalIn`	Use the HDMI input (Located to the left). "Primary Input" Z4K Dual input HDMI only. "HDMI IN2" on UHD60
`usbc`	Use the USB-C input.
`auto`	Use whatever port has an active input if only one source is connected. Note this is only valid for DisplayPort and Dual-HDMI options. Does not work with SDI or Analog inputs. Please see ZyPer4K User Guide for details on what port is used if both ports have an active input.
`displayPort`	Use the Display-Port input
`hdsdi`	Use the SDI input port
`12gsdi`	Use the 12G SDI input port
`component`	Use component input. (Requires ZeeVee Hydra cable)
`composite`	Use composite input (Requires ZeeVee Hydra cable)
`s-video`	Use s-video input (Audio not supported)
`vga`	Use vga input. (Requires ZeeVee VGA cable)

Example

```
set device myEncoder1 videoPort displayPort
Success
```

set encoder analogAudioOut

Sets the analog audio output source type for the specified encoder. (ZyPer4K and ZyPerUHD60 family only)
Also used to configure the port as an input.

Syntax

```
set encoder id analgoAudioOut source type
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

type

Type: STRING

The audio mode (analog or HDMI).

argument	Description
`none`	No analog audio output from the encoder. Port is configured for analgo audio input in this case.
`hdmiAudioDownmix`	Uses downmixed audio from input HDMI stream
`directDanteAudio`	Uses Dante audio that has been routed from another Dante Source to the encoder. (Only valid if Encoder as supporting Dante hardware installed) ZyPer4K-XSE Family only

Examples

```
set encoder Myencoder1 analogAudioOut source hdmiAudioDownmix
Success
set encoder Myencoder1 analogAudioOut source none
Success
set encoder MyDanteEncoder analogAudioOut source directDanteAudio
Success
```

set encoder danteAudioOut

Sets the audio input source source type for the specified encoder. (ZyPerUHD60 Dante Enabled Encoders only)

Only used when UHD60 Encoder is configured as a Dante Tranmitter

Syntax

```
set encoder id danteAudioOut source mode
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

mode

Type: STRING

The audio output to use.

argument	Description
`analogAudio`	Place the input Analog Audio from 5-pin phoenix connector onto Dante Network
`hdmiAudioDownmix`	Place the Downmix HDMI audio from HDMI connector onto Dante Network

Example

```
set encoder Myencoder1 danteAudioOut source hdmiAudioDownmix
Success
set encoder Myencoder1 danteAudioOut source analogAudio
Success
```

Related Commands

`set decoder danteAudioOut`

set encoder edid audio

Sets the allowable audio input formats at the encoder.

Detailed Background

ZeeVee added a feature that will allow compressed formats to be passed down in an encoder EDID file. This EDID will be then forwarded to the source device to determine the type of audio sent to the encoder.

This enhancement was to provide fast-switched connections the "compressed audio" options in the EDID file. Prior to this version with the fast-switched connection, ZeeVee modified the EDID passed from the decoder to the encoder and removed all compression formats. This left just LPCM as the only option under the "Audio data block" in the edid file.

```
>>> Audio data block <<<
  Linear PCM, max channels 8
    Supported sample rates (kHz): 192 176.4 96 88.2 48 44.1 32
    Supported sample sizes (bits): 24 20 16
```

The information provided to the Video Source device (such as BluRay Player or Media player) increases the possibility of compression being a chosen audio format. However it is still up to the device to choose uncompressed or compressed formats. It is important to know that some devices such as the Apple 4K TV requires the audio output type to be set (even if the audio format is available in the EDID). Compression will need to be set manually on these types of devices.

In addition any downmixed stream internal to ZyPer devices will not process compressed audio, so you will not hear compressed audio on these connections.

Syntax

`set encoder` *`id`* `edid audio` *`mode`*

Parameters

`id`

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

mode

Type: STRING

The supported input audio mode

argument	Description
`onlyPcm`	Force PCM audio format at encoder. Does not allow compressed formats such as AC3.
`allowCompressed`	Passes the decoders edid with unmodified audio information and thus allows compression options to be seen.
`serverDefault`	Follows the server setting

Example

```
set encoder Cuba edid audio allowCompressed
Success
```

Related Commands

`set server encoderDefaultAudioFormat`

Additional Information

In an attempt to properly Identify the Audio Streams used under the product the following changes were also made along with some modification to the API commands.

Product	Old Audio Stream Name	New Stream Name
ZyPer4K	hdmi (used in genlocked mode)	hdmiPassthroughAudio
ZyPer4K	hdmi-audio-downmix	hdmiAudio
ZyPer4K	analog-audio	analogAudio
ZyPerUHD	audio	hdmiAudio
ZyPerUHD	analog-audio	analogAudio
ZyPerUHD60	none	hdmiAudio
ZyPerUHD60	none	analogAudio

The ZyPer4k can have analog and digital audio streams going to the decoder at the same time and routing either way.

So:

```
join <enc> <dec> hdmiAudio
```

is simply used to route ‘standard’ HDMI audio from encoder to decoder.

Which port it goes out is based on defaults or the set command.

```
Zyper$ set decoder z4k_dec_desk_58 analogAudioOut source
    analogAudio
    hdmiAudioDownmix
```

or

```
Zyper$ set decoder z4k_dec_desk_58 hdmiAudioOut source
    analogAudio
    hdmiAudio (For HDMI out only)
    hdmiAudioDownmix
    hdmiPassthroughAudio (This is for genlockonly)
```

set encoder hdcpMode

Sets the hdcp mode for the specified encoder.

Syntax

```
set encoder id mode type
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

mode

Type: STRING

The hdcp mode to use

argument	Description
hdcpMode	HDCP mode of the Encoder.

type

Type: STRING

Enable or Disable

argument	Description
enabled	encoder will accept HDCP 1.4/2.2 compatible streams. Also will accept unencrypted inputs.
enabled1_4	encoder will accept HDCP 1.4 compatible streams. Also will accept unencrypted inputs.
disabled	encoder will reject HDCP 1.4/2.2 compatible streams. Will only accept unencrypted inputs.

Example

```
set encoder Myencoder1 hdcpMode disabled
Success
```

Notes

Useful when user does not want Source such as Apple Macbook to provide HDCP protected content to the Encoder.

set kvm

Configure kvm parameters. Decoders, Encoders, Multiviews and USB Source. (ZyPer4K-XS/XSE/XR only)

Syntax

```
set kvm name row <int> column <int> arg id
```

Parameters

name

Type: STRING

The name the kvm. String names are case sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
decoder	You are specifying a decoder for the kvm setup
encoder	You are specifying an encoder for the kvm setup
multiview	You are specifying an existing multiview for the kvm setup
usbSource	You are specifying the usb source for the kvm setup

id

Type: STRING

The name the decoder, encoder or multiview. String names are case sensitive.

Examples

```
set kvm ControlRoom row 1 col 1 decoder LargeSony
Success

set kvm ControlRoom row 2 col 1 multiview 2x2Grid
Success

set kvm ControlRoom row 1 col 1 usbSource
Success
```

set kvm hotkey

Sets the hotkey "base" for all kvm. (ZyPer4K-XS/XSE/XR only)

Syntax

```
set kvm all hotkeyBase option
```

Parameters

option

Type: STRING

Supply one of the following arguments.

argument	Description
`ctrl-ctrl`	Double Control Key activates Hotkey
`shift-shift`	Double Shift Key activates Hotkey
`alt-alt`	Double ALT Key activates Hotkey
`scroll-scroll`	Double SCROLL Key activates Hotkey
`print-print`	Double PRINT Key activates Hotkey

Example

```
set kvm all hotkeyBase alt-alt
Success

set kvm all hotkeyBase shift-shift
Success
```

How to Use

To activate a hot key function you first press the base keys. Example ctrl-ctrl. Then you press the corresponding key for the action you want to take

Key: S Action: Swap the window the mouse is in with the home position
Key: R Action: Reset all windows the settings in the saved configuration

Related Commands

```
create kvm
delete kvm
show kvm
start kvm
stop kvm
```

set ldap mode

Enables or disables Lightweight Directory Access Protocol (LDAP) mode.

Syntax

```
set ldap mode mode
```

Parameters

mode

Type: STRING

Supply one of the following arguments.

argument	Description
`disabled`	Disables ldap mode.
`enabled`	Enables ldap mode.

Example

```
set ldap mode disabled
Success
```

Related Commands

```
add ldap
delete ldap
show ldap
set ldap server address
```

set ldap server address

Sets the Lightweight Directory Access Protocol (LDAP) server domain.

Syntax

```
set ldap server address domain
```

Parameters

domain

Type: STRING

Domain of the LDAP server.

Example

```
set ldap server address zeevee.com
Success
```

Related Commands

```
add ldap
delete ldap
show ldap
set ldap server address
```

set ldap server baseSearchDn

Sets the Lightweight Directory Access Protocol (LDAP) server base search distinguished name.

The Base DN (also called Search Base) defines the location within the directory hierarchy where LDAP searches should start. It limits the search scope to a specific branch of the LDAP directory tree.

Syntax

```
set ldap server baseSearchDn name
```

Parameters

name

Type: STRING

base Dn of the LDAP server.

Example

```
set ldap server baseSearchDn employees
Success
```

Related Commands

```
add ldap
delete ldap
show ldap
set ldap server address
```

set ldap server bindDn username

Sets the Lightweight Directory Access Protocol (LDAP) server username.

The bindDN username in LDAP is the Distinguished Name (DN) of the account used by an application or client to authenticate (bind) to the LDAP server.

Used to log in to the LDAP server so the application can search or modify directory entries.

Often paired with a password for authentication.

Syntax

```
set ldap server bindDn username
```

Parameters

`username`

Type: STRING

username of the LDAP server.

Example

```
set ldap server bindDn username uid=jdoe,ou=people,dc=demo,dc=org
Success
```

Related Commands

```
add ldap
delete ldap
show ldap
set ldap server address
```

set ldap server bindDn password

Sets the Lightweight Directory Access Protocol (LDAP) server password.

Associated with the bindDN username in LDAP is the Distinguished Name (DN) of the account used by an application or client to authenticate (bind) to the LDAP server.

Used to log in to the LDAP server so the application can search or modify directory entries.

Syntax

```
set ldap server bindDn password
```

Parameters

password

Type: STRING

password of the LDAP server.

Example

```
set ldap server bindDn password admin123
Success
```

Related Commands

add ldap
delete ldap
show ldap
set ldap server address

set ldap type

Set the Lightweight Directory Access Protocol (LDAP) type to either openLDAP or ActiveDirectory

Syntax

```
set ldap type type
```

Parameters

type

Type: STRING

Supply one of the following arguments.

argument	Description
`openLdap`	Sets to openLdap.
`activeDirectory`	Sets to ActiveDirectory.

Example

```
set ldap type openLdap
Success
```

Related Commands

```
add ldap
delete ldap
show ldap
set ldap server address
```

set multiview

Assigns source to a position and size within a multiview display. (ZyPer4K family only)

Syntax

```
set multiview id windowNumber wn encoderName enc position
percentPositionX posx percentPositionY posy percentSizeX sx
percentSizeY sy layer ly
```

```
set multiview id windowNumber wn encoderName enc position
pixelPositionX posx pixelPositionY posy pixelSizeX sx pixelSizeY sy
layer ly
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

wn

Type: Integer

Window number within the multiview (1-19)

enc

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

percentPositionX

Type: Integer

X coordinate in percentage of multiview canvas. Upper left corner of window. (0-99)

percentPositionY

Type: Integer

Y coordinate in percentage of multiview canvas. Upper left corner of window. (0-99)

pixelPositionX

Type: Integer

X coordinate of multiview in multiview canvas. Upper left corner of window.

pixelPositionY

Type: Integer

Y coordinate of multiview in multiview canvas. Upper left corner of window.

`pixelSizeX`
Type: Integer

Size/Length of multiview window. Number of pixels in multiview canvas.

`pixelSizeY`
Type: Integer

Size/Height of multiview window. Number of Pixels in multiview canvas

`percentSizeX`
Type: Integer

Size/Length of multiview window. As a percentage of X dimension of multiview canvas. (0-99)

`percentSizeY`
Type: Integer

Size/Height of multiview window. As a percentage of Y dimension of multiview canvas. (0-99)

`ly`
Type: Integer

Window Layer. Value from 1-9 with layer 1 being the bottom layer and 9 being the top.

Examples

Using Percentages

```
set multiview myMview1 windowNumber 1 encoderName myEnc1
percentPositionX 50 percentPositionY 50 percentSizeX 25
percentSizeY 25 layer 3
```

Using Pixel Values

```
set multiview myMview1 windowNumber 1 encoderName myEnc1
pixelPositionX 1920 pixelPositionY 1080 pixelSizeX 800 pixelSizeY
600 layer 3
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
set multiview audioSource windowNumber
show multiviews config
show multiviews status
```

set multiview (layer, position, size)

Allows user to change a multiview window layer, position or size without specifying other parameters. (ZyPer4K family only)

Syntax

```
set multiview id windowNumber wn positionX posx positionY posy
sizeX sx sizeY sy layer ly
```

Parameters

`id`

Type: STRING

Name of previously created multiview. String names are case-sensitive.

`wn`

Type: Integer

Window number within the multiview (1-19)

`percentPositionX`

Type: Integer

X coordinate in percentage of multiview canvas. Upper left corner of window. (0-99)

`percentPositionY`

Type: Integer

Y coordinate in percentage of multiview canvas. Upper left corner of window. (0-99)

`pixelPositionX`

Type: Integer

X coordinate of multiview in multiview canvas. Upper left corner of window.

`pixelPositionY`

Type: Integer

Y coordinate of multiview in multiview canvas. Upper left corner of window.

`pixelSizeX`

Type: Integer

Size/Length of multiview window. Number of pixels in multiview canvas.

pixelSizeY
Type: Integer

Size/Height of multiview window. Number of Pixels in multiview canvas

percentSizeX
Type: Integer

Size/Length of multiview window. As a percentage of X dimension of multiview canvas. (0-99)

percentSizeY
Type: Integer

Size/Height of multiview window. As a percentage of Y dimension of multiview canvas. (0-99)

ly
Type: Integer

Window Layer. Value from 1-9 with layer 1 being the bottom layer and 9 being the top.

Examples

```
set multiview myMview1 windowNumber 2 layer 4
Success

set multiview myMview1 windowNumber 2 size percentSizeX 50 percentSizeY
50
Success

set multiview mv1 windowNumber 1 size pixelSizeX 500 pixelSizeY 400
Success
```

Related Commands

create multiview
delete multiview
delete multiviewWindow
set multiview audioSource windowNumber
show multiviews config
show multiviews status

set multiview allowMainStream

Controls if the main unscaled video stream from an encoder can be used in a multiview.
(ZyPer4K family only)

Syntax

```
set multiview id allowMainStream arg
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled (default)`	Unscaled stream is allowed
`disabled`	Unscaled stream is not allowed

Example

```
set multiview myMview1 allowMainStream enabled
Success
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
show multiviews status
```

set multiview audioSource windowNumber

Selects the input source to provide Audio for multiview display. (ZyPer4K family only)

Syntax

`set multiview id audioSource windowNumber arg`

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

arg

Type: STRING / Integer

Supply one of the following arguments.

argument	Description
`Integer`	Integer from 1-19 identifying source to use for audio
`none`	Set no audio for the multiview

Example

```
set multiview myMview1 audioSource window number 4
Success
```

Related Commands

`create multiview`
`delete multiview`
`delete multiviewWindow`
`show multiviews config`
`show multiviews status`

set multiview windowNumber channel up/down

Cycles the encoder source up/down for a specified multiview window. (ZyPer4K family only)

Syntax

```
set multiview id windowNumber channel arg
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

arg

Type: STRING

Supply one of the following arguments.

argument	Description
up	Will cycle the encoder source in the speficied multiview window to next higher numbered encoder. Will cycle around to lowest encoder number when maximum value is reached.
down	Will cycle the encoder source in the speficied multiview window to next lower numbered encoder. Will cycle around to highest encoder number when minimum value is reached.

Examples

```
set multiview mv3x3 windowNumber 3 channel up
Channel changed to Z4Kenc_2
Success

set multiview mv3x3 windowNumber 3 channel up
Channel changed to Arts_Encoder_1
Success
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
show multiviews status
```

set multiview canvasSize

Selects the canvas size for creating multiview windows. (ZyPer4K family only)

Helpful feature to control bandwidth of scaled streams for a multiview. Default canvas size is 3840x2160. This can create case where datarate from encoder is greater than 9.5Gb limit.(Full size stream plus scaled stream.) Reducing the canvas size will reduce required size and datarate of scaled stream used for multiview.

Syntax

```
set multiview id canvasSize pixelsHoriz pixelVert
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

pixelsHorz

Type: Integer

Horizontal width of the multiview canvas. (640 to 8192)

pixelsVert

Type: Integer

Vertical height of multiview window. (480 to 8192)

Note: Maximum canvas pixels is 8,847,360

Example

```
set multiview MyView1 canvasSize 1920 1080
Success
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
show multiviews status
```

set multiview newEncoderName

Assigns a new encoder to an existing multiview window. (ZyPer4K family only)

Syntax

```
set multiview id windowNumber wn newEncoderName encName|none
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

wn

Type: Integer

Window number within existing multiview. (1 to 19)

encName

Type: STRING / STRING

The name or MAC address of the encoder. String names are case sensitive.
None is also an option to remove existing encoder and replace with nothing.

Example

```
set multiview mv2x2-Art windowNumber 3 newEncoderName ABC
Success

set multiview mv2x2-Art windowNumber 3 newEncoderName none
Success
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
show multiviews status
```

set multiview title

Used to create a text overlay in a multiview window. (ZyPer4K family only)

Important note: This feature is not supported on ZyPer4K-XS, ZyPer4K-XR or ZyPer4K-XSE encoders. The ZyPer4K decoder is not relevant. All versions supported.

Create a string of text to be overlayed somewhere in a multiview window. Color of text and color of background can be specified. Size of text can be specified. Transparency of text and background can be specified. Note that 100% transparent setting is not fully transparent.

Syntax

```
set multiview id windowNumber wn title textString title

set multiview id windowNumber wn title text-size ts

set multiview id windowNumber wn title transparency text tt
background bt

set multiview id windowNumber wn title color text tc background bc
```

Parameters

id

Type: STRING

Name of previously created multiview. String names are case-sensitive.

wn

Type: Integer

Window number within the multiview (1-19)

ts

Type: Integer

Size of text (1-10)

tt

Type: Integer

Text Transparency. Percentage (0-100)

bt

Type: Integer

Background Transparency. Percentage (0-100)

tc

Type: STRING

Text color. Can be any of the following options: black, blue, brown, cyan, darkBlue, gray, green, lightBlue, lightGray, lime, magenta, maroon, olive, orange, purple, red, silver, white, yellow.

bc

Type: STRING

Background color. Can be any of the following options: black, blue, brown, cyan, darkBlue, gray, green, lightBlue, lightGray, lime, magenta, maroon, olive, orange, purple, red, silver, white, yellow.

title

Type: STRING

Any text string to be associted and displayed in the selected multiview window. Strings contains spaces must be enclosed in quotations.

Examples

```
set multiview MyView1 windowNumber 1 title textString "Window #1"
Success

set multiview MyView1 windowNumber 1 title textSize 10
Success

set multiview MyView1 windowNumber 1 title transparency text 0
background 100
Success

set multiview MyView1 windowNumber 1 title color black background-
color green
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
show multiviews status
```

set preset commands auto

Used to update an existing preset

Syntax

```
set preset id commands auto connections
```

Parameters

id

Type: STRING

The name of the preset. String names are case-sensitive.

connections

Supply one of the following arguments.

argument	Description
`existingConnections`	Uses the existing set of connections to create the command list
`empty`	Creates an empty set of commands. No connections

Example

```
set preset morning commands auto existingConnections
Success

set preset morning commands auto empty
Success
```

Related Commands

```
create preset
delete preset
run preset
show preset
```

set preset commands blob

Used to update an existing preset commands

Syntax

```
set preset id commands blob connections
```

Parameters

id

Type: STRING

The name of the preset. String names are case-sensitive.

connections

Type: STRING

Manually enter a list of commands contained within quotations. Insert a semi-colon between commands. Maximum character limit is 4096.

Example

```
set preset morning commands blob "join Cuba Bot-Left fast-
switched;join NBC Bot_Right fast-switched;join Sports Top-Right
fast-switched;join Media Player Top_Left fast-switched"
```

Below is image from ZMP GUI showing these commands in the Preset window:

Commands:

```
join Cuba Bot-Left fast-switched
join NBC Bot_Right fast-switched
join Sports Top-Right fast-switched
join MediaPlayer Top_Left fast-switched
```

Related Commands

create preset
delete preset
run preset
show preset

set preset description

Used to update an existing preset description

Syntax

```
set preset id description description
```

Parameters

id
Type: STRING

The name of the preset. String names are case-sensitive.

description
Type: STRING

Updated description of the preset

Example

```
set preset morning description "Open for business"
Success
```

Related Commands

create preset
delete preset
run preset
show preset

set preset schedule eventColor

Used to update an existing preset schedule color in the calendar.

Syntax

```
set preset id schedule scname eventColor color
```

Parameters

id

Type: STRING

The name of the preset. String names are case-sensitive.

scname

Type: STRING

Name of the schedule.

color

Type: STRING

Name of the new color. Options include the following: aqua, aquamarine, black, blue, brown, coral, cyan, darkBlue, darkSlateGray, deepPink, deepSkyBlue, fuchsia, gray, green, hotPink, khaki, lightBlue, lightGray, lightSeaGreen, lightSlateGray, lime, magenta, maroon, mistyRose, olive, orange, pink, purple, red, silver, teal, web-hex-color starting with # (e.g. #22ffee), white, yellow, zvGreen, zvPurple

Example

```
set preset morning schedule opentime zvGreen
Success
```

Related Commands

create preset
delete preset
run preset
show preset

set preset schedule month

Used to update an existing preset schedule month/day/time to run

Syntax

```
set preset id schedule scname month month dayOfMonth day dayOfWeek
day hour hour minute minute
```

Parameters

id

Type: STRING

The name of the preset. String names are case-sensitive.

scname

Type: STRING

Name of the schedule.

month

Type: STRING

Months to run this preset: Options are all, jan, feb, mar, apr, may, jun, jul, aug, oct, nov, dec

dayOfMonth

Type: Integer

Days of the month to run this preset. Enter an integer date or "all"

dayOfWeek

Type: STRING

Days of week to run this preset: Options are all, sunday, monday, tuesday, wednesday, thursday, friday, saturday, weekday, weekend. (Note: Weekday = M-F, Weekend = Sat+Sun)

hour

Type: String

Hour to run this preset. Enter an integer time (24 hour format) or "all"

---

minute

Type: Integer

Enter the minute (0-59) for this preset to run.

Example

```
set preset test1 schedule LateLunch month all dayOfMonth all
dayOfWeek weekday hour 14 minute 30
Success
```

Related Commands

create preset
delete preset
run preset
show preset

set responses rs232TermChars

Specifies the termination character for an RS232 string. The default string is "`\n\r`". Any character in the termination string causes the response-string to terminate and be placed into the response-string ring buffer.

This string is optional. If it is not specfied, then the string is empty and each low-level response is handled as a separate response.

Syntax

```
set responses id chr
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

chr

Type: STRING

The specified string.

Example

```
set responses decoder2 rs232TermChars "\r"
Success
```

Related Commands

`set device rs232`

set role

Sets permisison levels for a specific role. Note that the role must have been previously created.

Syntax

`set role rolename` *`rolename`* `subsystem` *`subinfo`* `maxAccess` *`accessLevel`*

Parameters

`rolename`

Type: STRING

String names are case-sensitive.

`subinfo`

Type: STRING

Supply one of the following arguments

argument	Description
`account`	Abiltiy of this role to modify accounts
`all`	Set all priority fields with a single setting
`device`	Ability of this role to modify devices
`ldap`	Ability of this role to modify ldap
`log`	Abiltiy of this role to access logs
`multiview`	Ability of this role to modify/create multiviews
`netmap`	Abiltiy of this role to modify netmaps
`preset`	Ability of this role to modify/create presets
`role`	Ability of this role to modify/create other roles
`server`	Ability of this role to modify server settings
`snmpagent`	Ability of this role to modify/view snmp
`tls`	Ability of this role to modify/view tls settings
`videowall`	Ability of this role to modify/create videowalls
`zone`	Ability of this role to modify/create zones

accessLevel

Type: STRING

Supply one of the following arguments

argument	Description
admin	Full unlimited access/control
config	Ability to configure
join	Ability to join (only applies to certain items such as multiviews)
none	No permissions
view	Ability to view only

Examples

```
set role rolename admin subsystem all maxAccess admin
Success

set role rolename junior subsystem all maxAccess admin
Success

set role rolename junior subsystem account maxAccess config
Success

set role rolename junior subsystem role maxAccess view
Success

set role rolename junior subsystem videowall maxAccess none
Success
```

Related Commands

create role
delete role
show role

set server api lineWrap

Sets the number of characters the API will display in the Command Line Interface before wrapping to a new line.

Syntax

```
set server api lineWrap wrap
```

Parameters

wrap

Type: INTEGER

Integer value from 100 to 512

Example

```
set server api lineWrap 200
Success
```

set server autoEdidMode

Sets the EDID mode for the Management Platform. By default, Auto-EDID mode is enabled.

Syntax

```
set server autoEdidMode mode
```

Parameters

mode

Type: STRING

Supply one of the following arguments.

argument	Description
`disabled`	Disables auto-EDID mode.
`enabled`	Enables auto-EDID mode.

Example

```
set server autoEdidMode disabled
Success
```

Related Commands

`set server timezone`

set server contact

Assign a server contact name.

Syntax

```
set server contact name
```

Parameters

name

Type: STRING

The name of the contact (String length from 1 to 256 characters)
Note use of Quotes “” is name includes a space character.

Example

```
set server contact “Art Weeks”
Success

show server info
server(192.168.0.21);
  server.gen; hostname=zyper.local, serverType=NUC-E,
version=3.3.39589, previousVersion=3.2.39546, master=true,
location=Wiring closet, contact=Art Weeks
  server.gen; uptime=2d:20h:42m:32s, freeMem= 3.05GB,
sdvoeVersion=3.6.0.1, bootCount=20, serialNumber=ZZM1MC00001A
  server.gen; macAddress=00:0e:c6:8e:38:6f, managementMacAddress=54
:b2:03:f0:ee:99
  server.ipActive; ipServerAddr=192.168.0.21,
ipManagementAddr=0.0.0.0, gatewayAddr=0.0.0.0, dnsAddr=0.0.0.0
  server.ipActive; managementGatewayAddr=0.0.0.0,
managementDnsAddr=0.0.0.0
  server.time; time=”Mon Jan  8 10:03:11 2024”, timezone=America/
New_York
  server.license; productID=367d8850-f6f2-cd73-fbb1-54b203f0ee99,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=2, allDevices=10,
allDevicesUp=9, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_1.7.24.0=2, num_1.7.26.0=1,
num_1.7.31.0=1, num_2.0.0.0=2, num_2.0.4.18=1, num_4.1.2.9=2
lastChangeIdMax(964);
```

set server dataTunnelMode

Sets the transfer mode for the Management Platform.

Syntax

```
set server dataTunnelMode mode
```

Parameters

mode

Type: STRING

Supply one of the following arguments.

argument	Description
`raw`	Sets raw communication mode.
`telnet`	Sets telnet communication mode.

Notes

Telnet is a way of passing control information about the communication channel.
It defines line-buffering, character echo, etc, and is done through a series of will/wont/do/dont messages when the connection starts.

Raw is a TCP stream with no telnet escape sequences.

Telnet is an application layer protocol while TCP is a transport layer protocol. Telnet uses TCP in order to transmit data. That is a big fundamental difference between Telnet and TCP.

Example

```
set server dataTunnelMode telnet
Success
```

Related Commands

`set server timezone`

set server date

Used to set server date manually or via ntp server. Note: NTP Server must be IPV4

Syntax

```
set server date mode

set server date ntpServer address <domainName>

set server date manual month <int> day <int> year <int> hour <int>
minute <int>
```

Parameters

mode

Type: STRING

Supply one of the following arguments.

argument	Description
`manual`	Sets date/time manually
`ntpServer`	Sets date/time via ntp server. Must provide valid IP address for an ntp Server.

Example

```
set server date manual month 4 day 1 year 2021 hour 15 minute 1
Success

set server date ntpServer address 129.6.15.28
Success
```

Link to NTP Servers:

https://tf.nist.gov/tf-cgi/servers.cgi

Related Commands

```
set server timezone
show server config
show server info
```

set server discoverMode

Sets how ZyPer endpoints are discovered by the Management Server on the network

Syntax

```
set server discoverMode all mode
```

Parameters

mode

Type: STRING

Supply one of the following arguments.

argument	Description
broadcast	Sets discovery mode to broadcast (Default)
multicast	Sets discovery mode to multicast
none	Will not learn devices (ZyPerUHD/UHD60 only)

Notes

Allows the server to discover ZyPer endpoints using multicast across subnets when multicast routing is enabled. When in multicast mode there must be a igmp querier running – usually that would be the multicast router querier.

Example

```
set server discoverMode all multicast
Success
```

Related Commands

set server timezone

set server encoderDefault audio

Sets the default encoder audio format for HDMI audio input.

Detailed Background

ZeeVee added a feature that will allow compressed formats to be passed down in an encoder EDID file. This EDID will be then forwarded to the source device to determine the type of audio sent to the encoder.

This enhancement was to provide fastSwitched connections the "compressed audio" options in the EDID file. Prior to this version with the fast-switched connection, ZeeVee modified the EDID passed from the decoder to the encoder and removed all compression formats. This left just LPCM as the only option under the "Audio data block" in the edid file.

```
>>> Audio data block <<<
  Linear PCM, max channels 8
    Supported sample rates (kHz): 192 176.4 96 88.2 48 44.1 32
    Supported sample sizes (bits): 24 20 16
```

The information provided to the Video Source device (such as BluRay Player or Media player) increases the possibility of compression being a chosen audio format.
However it is still up to the device to choose uncompressed or compressed formats.
It is important to know that some devices such as the Apple 4K TV requires the audio output type to be set (even if the audio format is available in the EDID). Compression will need to be set manually on these types of devices.

In addition any downmixed stream internal to ZyPer devices will not process compressed audio, so you will not hear compressed audio on these connections.

Syntax

`set server encoderDefault edid audio` *`mode`*

Parameters

`mode`

Type: STRING

Supply one of the following arguments.

argument	Description
`allowCompressed`	Passes the decoders edid with unmodified audio information and thus allows compression options to be seen.
`onlyPcm`	Forces the EDID modification described above

Example

```
set server encoderDefault edid audio allowCompressed
Success
```

Related Commands

`set encoder edid audio`

Additional Information

In an attempt to properly Identify the Audio Streams used under the product the following changes were also made along with some modification to the API commands.

Product	Old Audio Stream Name	New Stream Name
ZyPer4K	hdmi (used in genlocked mode)	hdmiPassthroughAudio
ZyPer4K	hdmi-audio-downmix	hdmiAudio
ZyPer4K	analog-audio	analogAudio
ZyPerUHD	audio	hdmiAudio
ZyPerUHD	analog-audio	analogAudio
ZyPerUHD60	none	hdmiAudio
ZyPerUHD60	none	analgoAudio

set server ftp mode

Used to enable to disable FTP access to the Management Server.

Syntax

```
set server ftp mode arg
```

Parameters

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Enable FTP access to the Management Server
`disabled`	Disable FTP access to the Management Server

Examples

```
set server ftp mode enabled
Success

set server ftp mode disabled
Success
```

Related Commands

`set server timezone`

set server hostname

Assign a server hostname. (Default is zyper.local)

Syntax

```
set server hostname name
```

Parameters

name

Type: STRING

The updated hostname (String length from 1 to 128 characters)
Note use of Quotes “” is name includes a space character.

Example

```
set server hostName myZMPserver
Success

show server info
server(192.168.0.21);
  server.gen; hostname=myZMPserver, serverType=NUC-E,
version=3.3.39589, previousVersion=3.2.39546, master=true,
location=Wiring closet, contact=Art Weeks
  server.gen; uptime=2d:20h:49m:51s, freeMem= 3.04GB,
sdvoeVersion=3.6.0.1, bootCount=20, serialNumber=ZZM1MC00001A
  server.gen; macAddress=00:0e:c6:8e:38:6f, managementMacAddress=54
:b2:03:f0:ee:99
  server.ipActive; ipServerAddr=192.168.0.21,
ipManagementAddr=0.0.0.0, gatewayAddr=0.0.0.0, dnsAddr=0.0.0.0
  server.ipActive; managementGatewayAddr=0.0.0.0,
managementDnsAddr=0.0.0.0
  server.time; time=”Mon Jan  8 10:10:30 2024”, timezone=America/
New_York
  server.license; productID=367d8850-f6f2-cd73-fbb1-54b203f0ee99,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=2, allDevices=10,
allDevicesUp=9, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_1.7.24.0=2, num_1.7.26.0=1,
num_1.7.31.0=1, num_2.0.0.0=2, num_2.0.4.18=1, num_4.1.2.9=2
lastChangeIdMax(968);
Success
```

set server ip

Sets the IP Address of the Management Platform. For MP hardware with multiple Network Interfaces this command is used to set the IP Address of each interface independently.

Syntax

```
set server ip id mode address IP Address mask Mask gateway Gateway
dns DNS-Server reboot
```

Parameters

id

Type: STRING

Supply one of the following arguments.

argument	Description
`videoPort`	Select the "Video" network. (ZyPer Network)
`managementPort`	Select the "Management" network. (Non-ZyPer Network)

mode

Type: STRING

Supply one of the following arguments.

argument	Description
`static`	Manually select/assign IP Address
`dhcp`	Allow DHCP server to automatically assign IP Address

Example

```
set server ip videoPo dhcp reboot
Success

set server ip videoPort static address 172.16.16.111 mask
255.255.255.0 gateway 172.16.6.1 dns none reboot
Success

set server ip managementPort static address 192.160.20.2 mask
255.255.255.0 gateway 192.168.20.1 dns none reboot
Success
```

Related Commands

`set server timezone`

set server isaac address

Sets the domain name of the isaac server.

Syntax

```
set server isaac address domainname
```

Parameters

domainname
Type: STRING

domainname of the Isaac server

Example

```
set server isaac address
Success
```

Related Commands

`set server isaac subsystemId`

set server isaac subsystemId

Sets the subsystemID on isaac server.

Syntax

```
set server isaac address subsystemId
```

Parameters

subsystemID
Type: STRING

Subsystem ID of the Isaac server

Example

```
set server isaac subsystemId Wallyworld
Success
```

Related Commands

`set server isaac address`

set server isolationMode

When in isolationMode, the server will not auto-discover other servers for redundancy (although it will keep the ones it already has discovered). And "set server discover mode none" will not discover devices (but will also keep the ones it already discovered). So... an existing site can transition to isolation mode very easily by using those two commands..

Syntax

```
set server isolationMode arg
```

Parameters

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Enable Isolation Mode
`disabled`	Disable Isolation Mode

Examples

```
set server isolationMode enabled
Warning:(75) Disabling discover mode presently only supported for
UHD devices
Warning:(84) Existing auto-discovered devices and MPs added to
manual-add list
Success


set server isolationMode disabled
Success
```

Related Commands

```
set server discoverMode
show server managedDevices
add device
delete device
```

Isolation Mode use cases

Case 1: Existing 2xMP (call them 1.1 and 1.2) in redundancy mode, not in isolation mode ==> change to isolation mode.

- Issue isolation mode from master
- Both MPs will be in isolation mode
- Both will only manage existing devices
- If a new device is added, must be manually added on the master
- Important: there is no functional change at the end of this process.
- The MPs still know about each other and all existing devices will be managed by them

Case 2: Existing MP1.1, MP1.2 in redundancy mode AND in isolation mode ==> add new single isolated MP-2.1

- Bring up MP-2.1 on physically separate network
- Set MP-2.1 to isolation mode
- Move MP-2.1 to active network along with MP1.1/1.2
- MP2.1 will not see MP1.1/MP1.2, and MP1.1/MP1.2 will not see MP2.1
- MP2.1 will not discover any devices
- Manually add devices to MP2.1 (ones not managed by MP1.1/MP1.2)

Case 3: Existing MP1.1/MP1.2 in redundancy mode and in isolation mode; MP2.1 in isolation mode ==> Add MP2.2 as redundant to MP2.1

- Bring up new MP2.2 on physically separate network
- Set MP2.2 to isolation mode
- Manually add MP2.1 to MP2.2
- Manually add MP2.2to MP2.1
- Move to MP2.2 to active network
- Strongly recommend shutting MP2.2 down, connecting to active network, bring up
- This will ensure it will become the slave, and the DB on MP2.1 will not be lost
- Now, there are two isolated MP redundant pairs!

set server license

Sets the license for the Management Platform. This controls the maximum number of endpoints supported by the Management Platform.

Syntax

```
set server license key
```

Parameters

key

Type: STRING

License key obtained from ZeeVee that sets maximum number of endpoints

Example

```
set server license QDZV-AYYA-0048-303D-5C0E-BD5D-56AA-154D-976C-
BCE3-BAC4
Success
```

Related Commands

`set server autoEdidMode`

set server location

Assign a server location.

Syntax

```
set server location name
```

Parameters

name

Type: STRING

The updated location (String length from 1 to 256 characters)
Note use of Quotes "" is name includes a space character.

Example

```
set server location "Art's Dining Room"
Success

show server info
server(192.168.0.21);
  server.gen; hostname=myZMPserver, serverType=NUC-E,
version=3.3.39589, previousVersion=3.2.39546, master=true,
location=Art's Dining Room, contact=Art Weeks
  server.gen; uptime=2d:20h:52m:56s, freeMem= 3.03GB,
sdvoeVersion=3.6.0.1, bootCount=20, serialNumber=ZZM1MC00001A
  server.gen; macAddress=00:0e:c6:8e:38:6f, managementMacAddress=54
:b2:03:f0:ee:99
  server.ipActive; ipServerAddr=192.168.0.21,
ipManagementAddr=0.0.0.0, gatewayAddr=0.0.0.0, dnsAddr=0.0.0.0
  server.ipActive; managementGatewayAddr=0.0.0.0,
managementDnsAddr=0.0.0.0
  server.time; time="Mon Jan  8 10:13:35 2024", timezone=America/
New_York
  server.license; productID=367d8850-f6f2-cd73-fbb1-54b203f0ee99,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=2, allDevices=10,
allDevicesUp=9, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_1.7.24.0=2, num_1.7.26.0=1,
num_1.7.31.0=1, num_2.0.0.0=2, num_2.0.4.18=1, num_4.1.2.9=2
lastChangeIdMax(970);
Success
```

set server redundancy mode

Used to enable or disable the reduncancy feature of a server when two servers are located in the same system.

Syntax

```
set server redundancy mode arg
```

Parameters

arg

Type: STRING

argument	Description
`enabled`	Enable redundancy feature (DEFAULT)
`disabled`	Disable redundancy feature

Note: The default setting for this feature is Enabled. Redundancy commands can be ignored in systems using only a single ZMP Server.

Examples

```
set server redundancy mode enabled
Success

set server redundancy mode disabled
Success
```

set server redundancy

Sets a virtual IP address and Mask for the Master and Slave Management Platforms in the system. (See Appendix for additional Redundancy Configuration Instructions)

Syntax

```
set server redundancy serv_id virtualIp address IP_Address
networkInterface video|management
```

Parameters

serv_id

Type: STRING

The servers to apply Virtual-ID to.

argument	Description
`allServers`	All Management Platforms on the Network. (Master and Slave)
`thisServer`	The specific server (Master or Slave) currently logged into.
`server IP Address`	Manually enter IP address of a specific Management Platform. (Master or Slave)

IP_Address and Mask

Type: STRING

Virtual IP address with Subnet Mask

argument	Description
`IP Address`	Virtual IP address to use for designated servers: Example: 192.168.0.25
`networkInterface`	Selects either the Video or Management interface for MP units with Dual Network Interfaces

Note: The virtual address has to be accessible within the subnet already defined for the interface. So, if the "video network", aka the original interface has 172.6.2.22/24, then the virtual address has to be 172.16.2.xxx.

Examples

```
set server redundancy allServers virtualIp address 192.168.0.25
networkInterface video
Success

set server redundancy thisServer preferredMaster true
preferredSlave false
Success

set server redundancy 192.168.1.202 preferredMaster false
preferredSlave true
Success
```

set server security deviceSecurityKey

Part of the mechanism to enable security over Semtech's server-device communication. First, there has to be an overall key associated with the server (deviceSecurityKey). Then, each device has to enable the security. It's authentication and encryption. This only works with ZyPer4K-XS/XSE and ZyPer4K-XR devices. Once a device has been enabled for a specific server, it will not work with any server without the same key. Although redundancy automatically sets the same key on both servers. If the key is lost, devices have to be hardware factory defaulted.

Syntax

```
set server security deviceSecurityKey key
```

Parameters

key

Type: STRING

Server security key. Text from 8 to 64 characters in length

Example

```
set server security deviceSecurityKey patriotsrule
Success
```

Notes

To change the server key; all devices (encoder and decoders) must have the security feature disabled first. Then change the key and re-enable the security feature on the devices.

Related Commands

`set server device security`

set server telnet password

Sets the password for Telnet. If a password is not provided, then the current password will be deleted. In this case, no password prompt will be displayed.

By default Telnet has no password.

Syntax

```
set server telnet pass
```

Parameters

pass

Type: STRING

The desired password.

Example

```
set server telnet password biGB055
Success
```

Notes

To reset system to no telnet password:
FTP the empty file named "defaultPasswords" to the /files directory of the MP (no file extension)
Power cycle the MP within 1 minute, when it comes back the passwords will be defaulted.

This provides the very secure requirement of having physical access to the MP in order to reset the password.

Related Commands

```
set server autoEdidMode
set server telnet mode
set server timezone
```

set server telnet mode

Used to enable or disable telnet access to the server.

Syntax

```
set server telnet mode mode
```

Parameters

mode
Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Telnet access is enabled
`disabled`	Telnet access is disabled

Example

```
set server telnet mode disabled
Success
```

Example trying to access via Telnet once disabled

```
telnet 192.168.0.78
Trying 192.168.0.78...
telnet: connect to address 192.168.0.78: Connection refused
telnet: Unable to connect to remote host
```

Related Commands

```
set server autoEdidMode
set server telnet password
set server timezone
```

set server timezone

Sets the time zone for the Management Platform. The time zone must be specified in POSIX format.

Syntax

```
set server timezone zone
```

Parameters

zone

Type: STRING

The time zone in POSIX format.

Example

```
set server timezone America/New_York
Success
```

Link to list of POSIX format timezones:

https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Related Commands

```
set server autoEdidMode
set server date ntpServer address <domainName>
set server date manual month <int> day <int> year <int> hour <int>
minute <int>
show server info
show server config
```

set terminal output

Set terminal output options between normal and JSON format.

The web interface has always been "JSON encoded responses" (computer friendly). The major benefit for this is for a web app to easily process the response. The downside is that it's not at all "human friendly".

There are two output format options from the API. One over telnet and ssh that is human friendly, and one over http that's computer friendly.

This new command allows users to select the format of responses from the API.

Syntax

```
set terminal output normal|json echo yes|no prompt yes|no
```

Parameters

normal | json - allows user to select between these two options

echo - allows characters/commands to be seen while typing in telnet/ssh session

prompt - provides "ZyPer$" prompt as que for entering commands in telnet/ssh session

Examples

```
set terminal output normal echo yes prompt yes
Success
```

```
set terminal output json echo yes prompt yes
[   114]{"status":"Success","text":[],"errors":[],"warnings":[],"co
mmand":"set terminal output json echo yes prompt yes "}
```

```
set terminal output json echo no prompt yes
[   112]{"status":"Success","text":[],"errors":[],"warnings":[],"co
mmand":"set terminal output json echo no prompt yes"}
```

Warning

Removing echo feature from a normal Telnet or SSH session can be challenging as the user would no longer be able to see the commands being typed into the Telnet or SSH window. Copy/paste the first example above to return to normal operation.

set snmp netNode (v3)

Updates SNMP netNode for use with the ZeeVee Visualization, Analysis and Monitoring tool (VAM).

Syntax

```
set snmp netnode nameId version v3 authType auth username name
password pass
```

Parameters

nameId

Type: STRING or INTEGER

Name or NetNode ID Number of the NetNode

auth

Type: STRING

argument	Description
sha1	Use SHA-1 Hash
sha512	Use SHA-512 Hash

name

Type: STRING

Username

pass

Type: STRING

Password.

Example

```
set snmp netNode Arts_8x8 version v3 authType sha512 username admin
password netgear96
Success
```

Related Commands

```
delete snmp netNode
set snmp netNode
show snmpNode
```

set snmp netNode (v2c)

Updates SNMP netNode for use with the ZeeVee Visualization, Analysis and Monitoring tool (VAM).

Syntax

```
set snmp netnode nameID snmp v2c communityName name
```

Parameters

nameID

Type: STRING or INTEGER

Name or NetNode ID Number of the NetNode

name

Type: STRING

Community Name

Example

```
set snmp netNode myZMPserver version v2c communityName ZeeVee
Success
```

Related Commands

delete snmp netNode
set snmp netNode
show snmpNode

set snmp netNode ipAddress

Used to update or change ipAddress of SNMP netNode for use with the ZeeVee Visualization, Analysis and Monitoring tool (VAM).

Syntax

```
set snmp netnode ipaddr ipAddress newipaddr
```

Parameters

ipaddr

Type: STRING

Current IP Address of the SNMP NetNode

newipaddr

Type: STRING

New/Updated IP Addressof the SNMP NetNode

Example

```
set snmp netNode 192.168.0.222 ipAddress 192.168.0.1
Success
```

Related Commands

```
delete snmp netNode
set snmp netNode
show snmpNode
```

set tls server mode

Used to enable web server TLS mode.

Syntax

```
set tls server mode mode
```

Parameters

mode
Type: STRING

Supply one of the following arguments.

argument	Description
`enabled`	Telnet access is enabled
`disabled`	Telnet access is disabled

Example

```
set tls server mode enabled
Success

set tls server mode disabled
Success
```

Related Commands

```
show tls summary
show tls pem ca signedCert
```

set tls server fqdn

Used to set server Fully Qualified Domain Name. Either manually entered by user or contained in Certification file.

Syntax

```
set tls server fqdn domain | fromCert
```

Parameters

domain

Type: STRING

Full domain name

Examples

```
set tls server fqdn www.zeevee.com
Success

set tls server fqdn fromCert
Success
```

Related Commands

```
show tls summary
show tls pem ca signedCert
```

set videoWall size

Changes the size of the specified video wall and bezel parameters. Bezel values are measured in pixels.

> Setting bezel values will affect a resolution change to the display. If the resolution is not supported by the display, then the display will have no picture. If this is the case, try assigning a different bezel pixel value.

Syntax

```
set videoWall id size rows rows columns cols topBezel bezt
bottomBezel bezb leftBezel bezl rightBezel bezr
```

Parameters

id

Type: STRING

The name of the video wall. String names are case-sensitive.

rows

Type: INTEGER

The number of rows. (Maximum 15 for ZyPer4K, ZyPerUHD or ZyPerUHD60. Maximum 4 for ZyPerHD)

cols

Type: INTEGER

The number of columns. (Max 15 for ZyPer4K, ZyPerUHD or ZyPerUHD60. Max 4 for ZyPerHD)

bezt

Type: INTEGER

The top bezel pixel value.

bezb

Type: INTEGER

The bottom bezel pixel value.

bezl

Type: INTEGER

The left bezel pixel value.

bezr

Type: INTEGER

The right bezel pixel value.

Note: Bezel adjustment only supported on ZyPer4K family

Example

```
set videoWall Mywall1 size rows 5 columns 5 topBezel 0 bottomBezel
0 leftBezel 0 rightBezel 0
Success
```

Related Commands

```
create videoWall
set videoWall decoder
show videoWalls
join videoWall
set videoWall newName
```

set videoWall decoder

Assigns the specified decoder, to the desired row and column, on the specified video wall.

Syntax

```
set videoWall wallid decoder id row col
```

Parameters

id

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive. If `none` is passed as the argument, then any existing display is disconnected from that position in the video wall.

wallid

Type: STRING

The name of the video wall. String names are case-sensitive.

row

Type: INTEGER

The row of the specified video wall.

col

Type: INTEGER

The column of the specified video wall.

Example

```
set videoWall myVideoWall decoder myDecoder row 2 column 3
Success
```

Related Commands

```
create videoWall
set videoWall size
show videoWalls
join videoWall
```

set videoWall newName

Changes the name of an existing video wall

Syntax

```
set videoWall id newName name
```

Parameters

id

Type: STRING or MAC Address

The name of the video wall. String names are case-sensitive.

name

Type: STRING

The updated name of the video wall.

Example

```
set videoWall myWall2 newName yourWall2
Success
```

Related Commands

create videoWall
set videoWall size
show videoWalls
join videoWall

show account

Displays information about accounts

Syntax

`show account` *`select [since]`*

Parameters

`select`

Type: STRING

Supply one of the following arguments.

argument	Description
`active users`	Used to show information about currently active users logged into the system
`allConfig`	Shows account settings that apply to all accounts
`list`	Shows information about accounts including security settings and status
`login banner filenames`	Shows currently used banner image filenames
`login banner text webPreLogin`	Shows currently used login banner Pre login text
`login banner text webPostLogin`	Shows currently used login banner Post login text.

`since`

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Examples

```
show account active users all
session(1);
  session.status; user=admin, type=Telnet, extId=none,
start=12/21/22T13:55:10-0500, lastActive=12/21/22T14:07:04-0500
session(2);
  session.status; user=admin, type=Web, extId=Qf,lebMNNAn,
start=12/21/22T14:00:10-0500, lastActive=12/21/22T14:04:30-0500
lastChangeIdMax(19478);
Success

show account allConfig
allAccounts(192.168.0.22);
  allAccounts.gen; idleLogoutMins=unlimited, concurrentSesionsMax=5
  allAccounts.password; complexity=disabled, minLen=NA,
initialExpire=disabled, minDays=0, maxDays=unlimited
  allAccounts.onThreeFailures; lockoutMins=none, disable=false
  allAccounts.authMode; telnetFullAuth=oldAuth, webFullAuth=noAuth
lastChangeIdMax(9);
Success

show account list all
account(admin);
  account.gen; role=admin, lastLogin=12/21/22T14:00:17-0500,
twoFactor=disabled
  account.status; locked=disabled, passwordExpires=never
account(zyper);
  account.gen; role=admin, lastLogin=none, twoFactor=disabled
  account.status; locked=disabled, passwordExpires=never
account(sftp);
  account.gen; role=none, lastLogin=none, twoFactor=disabled
  account.status; locked=disabled, passwordExpires=never
lastChangeIdMax(9);
Success

show account login banner filenames
allAccounts(192.168.0.22);
  allAccounts.webBanners; preLoginText=none,
postLoginText=securePre.txt, preLoginImage=none,
postLoginImage=mickey.png
  allAccounts.terminalBanners; preLoginText=securePre.txt,
postLoginText=securePost.txt
lastChangeIdMax(9);
Success

show account login banner text webPostLogin
allAccounts(192.168.0.22);
  allAccounts.bannerText; webPostLogin="You are about enter a
secure site.\nIf you do not have authorization, do not proceed."
lastChangeIdMax(17);
Success
```

show dataTunnels

Shows what rs232 or IR data relay ports are opened on the server.

The feature of data-relays was added to allow a third party to connect to the ZMP server with a specific port and pass raw or telnet API commands (depending on the mode) to the server and port which is designated for a particular encoder or decoder.

Syntax

```
show dataTunnels
```

Parameters

none

Example

```
show dataTunnels
dataSessions(d8:80:39:9a:96:7);
  device: name=Cuba
  irTunnel: port=1234
  irTunnel-connections: none
Success
```

Related Commands

dataConnect
set server dataTunnelMode

show device capabilities

Displays device capabilities for the specified device(s).

Syntax

`show device capabilities` *`id select [since]`*

Parameters

`id`

Type: STRING or MAC Address

The identifier of the device. Either the full or portion of a string name or MAC address can be supplied.

`select`

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays configuration information for all available devices.
`encoders`	Only encoders are displayed.
`decoders`	Only decoders are displayed.

`since`

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show device capabilities MyEnc1 since 20
device(d8:80:39:eb:1:cb);
  device.gen; lastChangeId=28
  device.CapabilitiesVersion; values=1
  device.analogAudioPort; values=none:hdmiAudioDownmix
  device.colorDepth; values=fastSwitchDeepColor:multiviewDeepColor
  device.colorEncoding; values=fastSwitchSubsample:multiviewSubsam
ple
  device.edid; values=save:load
  device.edidAudioFormat; values=onlyPcm:allowCompressed:serverDef
ault
  device.ethernetManagementPortMode; values=enabled:disabled
  device.factoryDefaults; values=supported
  device.firmwareUpdate; values=...apz
  device.flashLeds; values=supported
  device.hdcpMode; values=enabled:enabled1.4:disabled
  device.hdmiStatus; values=link:hdcp:resolution:fps
  device.ipMode; values=dhcp:static
  device.ipStaticGateway; values=supported
  device.ir; values=device:server:none
  device.joinAudio; values=analogAudio:hdmiAudio
  device.joinUsb; values=false
  device.joinVideo; values=fastSwitched:genlocked:genlockedScaled:m
ultiview:window
  device.multiview; values=title
  device.previewStream; values=enabled:disabled
  device.rs232; values=device:server:none
  device.sendMulticasts; values=settable
  device.streamMcastSettable; values=video:analogAudio:hdmiAudio
  device.streamModeSettable; values=video:hdmiAudio:videoScaled:ana
logAudio
  device.streamsSupported; values=video:hdmiAudio:videoScaled:analo
gAudio
  device.temperature; values=main
  device.usbFilter; values=none
  device.videoPort; values=hdmi:auto
  device.videoWall; values=maxSize(15):bezelsSupported
lastChangeIdMax(29);
Success
```

Related Commands

show device status
show device config

show device config

Displays device information for the specified device(s).

Syntax

`show device config` *`id [since]`*

Parameters

`id`

Type: STRING or MAC Address

The identifier of the device. Either the full or portion of a string name or MAC address can be supplied. Can also enter in one of the arguments below.show

argument	Description
`all`	Displays configuration information for all available devices.
`commands`	Shows all the commands used to configure every device, multiview, video wall in the system. (Can be a lot of output)
`encoders`	Only encoders are displayed.
`decoders`	Only decoders are displayed.

`since`

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show device config XSdec
device(0:16:c0:4d:e3:12);
  device.gen; model=Zyper4KXS, type=decoder, virtualType=none,
name=XSdec, state=Up, lastChangeId=173
  device.gen; productCode=Z4KDECFXS, productDescription=Fiber
Decoder - HDMI 2.0, pid=0xd
  device.gen; controlAuthenticationMode=disabled
  device.gen; firmware=1.3.2.4
  device.gen; ethernetManagementPortMode=disabled
  device.optionalPorts; video=none, usb=hid, analogAudio=yes,
rs232=no, ir=no
  device.hdmi; hdcpMode=auto, 5vControl=disabled
  device.ports; videoPort=auto
  device.ip; mode=dhcp, address=169.254.19.227, mask=255.255.0.0,
gateway=NA
  device.display; iconImageName=GenericDisplay, manufacturer=none,
model=none, location=none, serialNumber=none
  device.edid; preferMode=strict
  device.display; mode=stretch
  device.displayResolution; allParameters=auto
  device.displayTiming; allParameters=auto
  device.connectedEncoder; macAddr=0:16:c0:4d:e3:67, name=XSenc_1,
connectionMode=fastSwitched
  device.audioConnections; analogSourceMac=none,
analogSourceName=none, hdmiAudioSourceMac=0:16:c0:4d:e3:67,
hdmiAudioSourceName=XSenc_1
  device.autoAudioConnections; hdmiAudioFollowVideo=false
  device.audioOutSourceType; analogOutSourceType=hdmiAudioDownmix,
hdmiOutSourceType=hdmiAudio
  device.usb; filter=none, internalIpAddress=none
  device.usbUplink; macAddr=none, name=none
lastChangeIdMax(176);
lastDeleteIdMax(3);
Success
```

Related Commands

show device status
show device capabilities
show device connections

show device connections

Shows encoder connections to decoders

Syntax

```
show device connections
```

Parameters

none

Example

```
show device connections
encoder.GalapogosHD; BotLeftHD
encoder.RaptorsHD; SamsungHD
encoder.MuralsHD; BotRightHD
encoder.Soccer4K; TopRight, BotLeft
Success
```

Related Commands

show device status
show device capabilities
show device config

show device names

Displays status information for the specified device(s). This command functions the same as the `show device config` command.

Syntax

```
show device names
```

Parameters

id

Type: STRING or MAC Address

Supply one of the following arguments.

argument	Description
`all`	Displays configuration information for all available devices.
`encoders`	Only encoders are displayed.
`decoders`	Only decoders are displayed.

Example

```
show device names all
name=Z4K-XS-Dec, mac=00:16:c0:4d:e3:12, ip=192.168.0.12, state=Down
name=Z4K-Enc-XS, mac=00:16:c0:4d:e3:67, ip=192.168.0.3, state=Down
name=UHD60-1D, mac=00:1c:d5:01:09:65, ip=192.168.0.52, state=Down
name=UHD60-0EA, mac=00:1c:d5:01:0a:a3, ip=192.168.0.63, state=Down
name=UHD60-2EA, mac=00:1c:d5:01:13:bd, ip=192.168.0.75, state=Up
name=UHD60-1EA, mac=00:1c:d5:01:13:c0, ip=192.168.0.66, state=Up
name=UHD60-0E, mac=34:1b:22:81:f9:81, ip=192.168.0.61, state=Up
name=UHD60-1DA, mac=34:1b:22:81:f9:88, ip=192.168.0.54, state=Up
name=XSE-Enc-Combo, mac=6c:df:fb:00:03:9e, ip=192.168.0.14,
state=Up
name=XSE-Dec-Combo, mac=6c:df:fb:00:03:a0, ip=192.168.0.9, state=Up
name=XSE-Decoder, mac=6c:df:fb:00:03:c2, ip=192.168.0.8, state=Up
name=XSE-Encoder, mac=6c:df:fb:00:03:c6, ip=192.168.0.7, state=Up
name=Z4K-Dec2, mac=d8:80:39:59:f0:36, ip=192.168.0.6, state=Down
name=Z4K-Enc2, mac=d8:80:39:9a:95:2f, ip=192.168.0.7, state=Down
Success
```

Related Commands

`add device`
`show device status`
`show device capabilities`
`show device config`

show device status

Displays status information for the specified device(s). This command functions the same as the `show device config` command.

Syntax

`show device status` *`id [since]`*

Parameters

`id`

Type: STRING or MAC Address

The identifier of the device. Either the full or portion of a string name or MAC address can be supplied.

Supply one of the following arguments.

argument	Description
`all`	Displays configuration information for all available devices.
`encoders`	Only encoders are displayed.
`decoders`	Only decoders are displayed.

`since`

This parameter is optional and can be specified to display units based on the number of changes. Supply this argument followed by the desired value to query.

argument	Description
`since`	Required when using this parameter.

Example

```
show device status Cuba
device(d8:80:39:9a:96:7);
  device.gen; model=Zyper4K, type=encoder, name=Cuba, state=Up,
uptime=4d:1h:57m:24s, lastChangeId=78
  device.temperature; main=59C
  device.firmwareUpdate; status=idle, loadingFile=none, percentComplete=0
  device.hdmiInput; cableConnected=connected, hdcp=inactive,
hdcpVersion=none, hdmi2.0=yes, horizontalSize=1280, verticalSize=720,
fps=60.000, interlaced=no
  device.hdmiInput; hTot=1650, hBlank=370, hFront=110, hSync=40,
hSyncPol=positive
  device.hdmiInput; vTot=750, vBlank=30, vFront=5, vSync=5, vSyncPol=positive
  device.hdmiInput; pixelClock=74.250, colorEncoding=YCBCR_444, colorDepth=8,
colorSpace=BT709, colorQuantRange=limited, timingStandard=CEA-861-F VIC-4
  device.edid; sourceType=file, sourceFilename=George.edid
  device.edid; edidStatus=valid, edidMonitorName=SyncMaster
  device.edid; firstDescriptorPreferredResolution=yes
  device.edid; maxFps=75.00, maxPixelClockMhz=170.00,
maxDeepColorPixelClockMhz=0.00, rgbColorDepth=8, yuv420ColorDepth=0
  device.edid; only420=none, also420=none, yuvQuantRange=default,
rgbQuantRange=default
  device.edid.audio.PCM; channels=2, sampleRates=48Khz-44.1Khz-32Khz,
sampleBits=16-20
  device.edid.preferredResolution; pixelClockMhz=148.50, sizeX=1920,
sizeY=1080, fps=60.00
  device.edid.maxResolution; pixelClockMhz=148.50, sizeX=1920, sizeY=1080,
fps=60.00
  device.videoStream; inputFps=60.00, inputDatarate=1451Mbps,
compressionFactor= 1.00, streamFps=60.00, streamDatarate=1451Mbps
  device.videoScaledStream; inputFps=60.00, inputDatarate=1451Mbps,
streamFps=30.00, streamDatarate=0Mbps
  device.previewStream; status=down, recvData=false
lastChangeIdMax(78);
Success
```

Related Commands

`show device config`

show device userAdded

Shows devices that have been manually added to the Management Platform using the add device command.

Syntax

```
show device userAdded
```

Parameters

none

Example

```
show device userAdded
device(d8:80:39:eb:1c:ee);
device.gen; model=Zyper4K, type=encoder, name=London, state=Up,
uptime=0d:18h:32m:36s, lastChangeId=55
device.ip; address=192.168.10.79
device(d8:80:39:59:f1:ff);
device.gen; model=Zyper4K, type=decoder, name=Right, state=Up,
uptime=0d:18h:32m:36s, lastChangeId=52
device.ip; address=192.168.10.81
device(d8:80:39:59:af:be);
device.gen; model=Zyper4K, type=decoder, name=Left, state=Up,
uptime=0d:18h:30m:5s, lastChangeId=56
device.ip; address=192.168.10.82
Success
```

Related Commands

add device
show device status
show device capabilities
show device config

show files

Shows files currently stored on the Managment Server. (EDID, Firmware, Icons and Idle Images)

Syntax

```
show files type
```

Parameters

type

Type: STRING

Supply one of the following arguments.

argument	Description
all	Show all files. (EDID, Firmware, Icons and Idle Images)
edid	Show EDID files.
firmware	Show Firmware files.
icon	Show Icon files.
idleImage	Show Idle Image files (ZyPerUHD and ZyPerUHD60 use)

Examples

```
show files icon
server(192.168.0.22);
  files.encoderIcon; names=SatelliteReceiver.png:BluRay.png:ps3.
png:SecurityCamera.png:FlatPanelDisplay.png:Laptop.png:cbs.
png:BroadcastCamera.png:fox.png:abc.jpg:DVD.png:xbox.png:DesktopPC.
png:MediaPlayer.png:tennis.png:espn.png:VCR.png:cnn.jpg:nbc.
png:foxSports.png:CableBox.png:golf.png:nflNetwork.jpg
  files.decoderIcon; names=FlatPanelDisplay.png:Projector.png:vw.png
  files.savedIcon; names=none
lastChangeIdMax(1);
Success

show files idleImage
server(192.168.0.22);
  files.idleImage; names=001Rupdated.jpg:IPD5000-B70_idle_image_
v1.0.jpg:SLupdated.jpg:test720.jpg
lastChangeIdMax(1);
Success
```

show kvm config

Shows all kvms and complete configuration settings. (ZyPer4K-XS/XSE and XR only)

Syntax

```
show kvm config [since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes. Supply this argument followed by the desired value to query.

argument	Description
since	Required when using this parameter.

Example

```
show kvm config
kvm(all);
  kvm.gen; hotkeyBase=shift-shift, swapHomeHotkey=shift-shift-s,
resetHomeHotkey=shift-shift-r
kvm(2x2wall);
  kvm.gen; started=false
  kvm.gen; usbSourceRow=1, usbSourceCol=2, homePositionRow=1,
homePositionCol=2
  kvm.position.1.1; decoder=Top_Left, encoder=Fenway_5
  kvm.position.1.2; decoder=Top_Right, encoder=Canada_6
  kvm.position.2.1; decoder=Bot_Left, encoder=Airshow_7
  kvm.position.2.2; decoder=Bot_Right, encoder=London_8
kvm(Multiview);
  kvm.gen; started=false
  kvm.gen; usbSourceRow=1, usbSourceCol=1, homePositionRow=1,
homePositionCol=2
  kvm.position.1.1; decoder=Top_Left, multiview=kvmMV
  kvm.position.1.2; decoder=Top_Right, encoder=London_8
lastChangeIdMax(73);
Success
```

Related Commands

```
create kvm
delete kvm
set kvm
start kvm
stop kvm
```

show kvm swapped

Shows all kvms with encoder/decoder settings. (ZyPer4K-XS/XSE and XR only)

Syntax

```
show kvm swapped [since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes. Supply this argument followed by the desired value to query.

argument	Description
since	Required when using this parameter.

Example

```
show kvm swapped
kvm(2x2wall);
  kvm.gen; started=false
  kvm.position.1.1; decoder=Top_Left, encoder=Fenway_5
  kvm.position.1.2; decoder=Top_Right, encoder=Canada_6
  kvm.position.2.1; decoder=Bot_Left, encoder=Airshow_7
  kvm.position.2.2; decoder=Bot_Right, encoder=London_8
kvm(Multiview);
  kvm.gen; started=false
  kvm.position.1.1; decoder=Top_Left, multiview=kvmMV
  kvm.position.1.2; decoder=Top_Right, encoder=London_8
lastChangeIdMax(49);
Success
```

Related Commands

```
create kvm
delete kvm
set kvm
start kvm
stop kvm
```

show ldap

Shows LDAP server info or Group ID info.

Syntax

```
show ldap type
```

Parameters

type

Type: STRING

Supply one of the following arguments.

argument	Description
`server`	Show LDAP server information
`groupIdToRole`	Show LDAP ID's and associated Roles

Examples

```
show ldap groupIdToRole
ldapGroupId(1234);
  ldapGroupId.gen; role=Tech
ldapGroupId(4569);
  ldapGroupId.gen; role=admin
ldapGroupId(6789);
  ldapGroupId.gen; role=admin
Success

show ldap server
ldapServer(172.16.6.111);
  ldap.gen; mode=enabled, type=openLdap, status=updating
  ldap.gen; serverAddress=zeevee.com, baseSearchDn="employees"
  ldap.gen; bindDnUsername="uid=admin,ou=people,dc=zeevee,dc=com"
lastChangeIdMax(31);
Success
```

Related Commands

```
add ldap
delete ldap
set ldap server bindDn username
set ldap server bindDn password
```

show logs authentications

Shows a listing of server login/logout events ordered from newest to oldest.

Syntax

```
show logs authentications max quantity
```

Parameters

quantity
Type: INTEGER

Number of past authentications to display

Example

```
show logs authentications max 5
log(192.168.0.22);
  log.msg.1; dt=Dec-16-22-12:47:02, user=system, sid=0, msg="EVENT
for server; Login -- account=admin, sessionId=1"
  log.msg.2; dt=Dec-16-22-12:47:00, user=system, sid=0,
msg="EVENT for server; Logout -- account=admin, sessionId=1,
reason=remoteClose"
  log.msg.3; dt=Dec-16-22-12:46:43, user=system, sid=0, msg="EVENT
for server; Login -- account=admin, sessionId=1"
  log.msg.4; dt=Dec-16-22-12:46:40, user=system, sid=0,
msg="EVENT for server; Logout -- account=admin, sessionId=1,
reason=remoteClose"
  log.msg.5; dt=Dec-16-22-12:46:40, user=system, sid=0, msg="EVENT
for server; Logout -- account=admin, sessionId=1, reason=User"
Success
```

Related Commands

show logs commands

show logs commands

Shows a listing of last commands send to the Management Server.

Syntax

```
show logs commands max quantity
```

Parameters

quantity

Type: INTEGER

Number of past commands to display

Example

```
show logs commands max 5
log(192.168.0.22);
  log.msg.1; dt=Dec-16-22-12:43:38, user=admin, sid=1,
msg="CommandLine: show logs commands max 5"
  log.msg.2; dt=Dec-16-22-12:43:36, user=admin, sid=2,
msg="Error:(29) Device Z4KDante does not support or cannot change:
joinUsb with value true."
  log.msg.3; dt=Dec-16-22-12:43:36, user=admin, sid=2,
msg="CommandLine: join Encl Z4KDante usb"
  log.msg.4; dt=Dec-16-22-12:43:36, user=admin, sid=2,
msg="Error:(29) Device Z4KDante does not support or cannot change:
joinUsb with value true."
  log.msg.5; dt=Dec-16-22-12:43:36, user=admin, sid=2,
msg="CommandLine: join none Z4KDante usb"
Success
```

Related Commands

show logs authentications

show multiviews config

Shows configuration information on all multiview displays. (ZyPer4K family only)

Syntax

```
show multiviews config
```

Parameters

none

Example

```
show multiviews config
multiview(Ltest1);
  multiview.audio; sourceWindow=none;
  multiview.window1; encoder-name=Airshow4K, percentPosX=40,
percentPosY=5, percentSizeX=55, percentSizeY=55, layer=1;
  multiview.window2; encoder-name=Soccer4K, percentPosX=5,
percentPosY=5, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window3; encoder-name=Wildlife4K, percentPosX=5,
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window4; encoder-name=Soccer4K, percentPosX=65,
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window5; encoder-name=USA4K, percentPosX=5,
percentPosY=35, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window6; encoder-name=USA4K, percentPosX=35,
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
multiview(MView4k);
  multiview.audio; sourceWindow=1;
  multiview.window1; encoder-name=Airshow4K, percentPosX=0,
percentPosY=0, percentSizeX=50, percentSizeY=50, layer=1;
  multiview.window2; encoder-name=USA4K, percentPosX=0,
percentPosY=50, percentSizeX=50, percentSizeY=50, layer=1;
  multiview.window3; encoder-name=Soccer4K, percentPosX=50,
percentPosY=0, percentSizeX=50, percentSizeY=50, layer=1;
  multiview.window4; encoder-name=Wildlife4K, percentPosX=50,
percentPosY=50, percentSizeX=50, percentSizeY=50, layer=1;
multiview(LBar);
  multiview.audio; sourceWindow=none;
  multiview.window1; encoder-name=Soccer4K, percentPosX=5,
percentPosY=5, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window2; encoder-name=Wildlife4K, percentPosX=5,
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window3; encoder-name=USA4K, percentPosX=35,
```

```
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window4; encoder-name=Soccer4K, percentPosX=65,
percentPosY=65, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window5; encoder-name=USA4K, percentPosX=5,
percentPosY=35, percentSizeX=30, percentSizeY=30, layer=1;
  multiview.window6; encoder-name=Airshow4K, percentPosX=35,
percentPosY=5, percentSizeX=60, percentSizeY=60, layer=1;
Success
```

Related Commands

create multiview
delete multiview
delete multiviewWindow
show multiviews status

show multiviews status

Shows status information for all multiview displays. (ZyPer4K family only)

Syntax

```
show multiviews status
```

Parameters

none

Example

```
show multiviews status
multiview(mv1);
  multiview.gen; totalDatarate=0Mbps
  multiview.window1; encoderName=MediaPlayer,
encoderMac=d8:80:39:eb:1:cb, streamType=none, datarate=0Mbps,
multicast=0.0.0.0, titleStatus=none, status=inactive, reason=no
decoder joined
  multiview.window2; encoderName=Curved,
encoderMac=d8:80:39:9a:e6:d, streamType=none, datarate=0Mbps,
multicast=0.0.0.0, titleStatus=none, status=inactive, reason=no
decoder joined
  multiview.window3; encoderName=Cuba,
encoderMac=d8:80:39:9a:96:7, streamType=none, datarate=0Mbps,
multicast=0.0.0.0, titleStatus=none, status=inactive, reason=no
decoder joined
  multiview.window4; encoderName=Camera2,
encoderMac=d8:80:39:9a:af:a3, streamType=none, datarate=0Mbps,
multicast=111.117.114.99, titleStatus=none, status=inactive,
reason=no decoder joined
Success
```

Related Commands

```
create multiview
delete multiview
delete multiviewWindow
show multiviews config
```

show multiviews titles

Shows title information for all multiview displays. (ZyPer4K family only)

Syntax

```
show multiviews titles arg
```

Parameters

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`config`	Displays title configuration information for multiview.
`text`	Displays text configuration information for multiview.

Examples

```
show multiviews titles text
multiview(mv1);
  multiview.gen; audioSourceWindow=none, canvasWidth=3840,
canvasHeight=2160
  multiview.window1; title=Window1
  multiview.window2; title=Window2
  multiview.window3; title=none
  multiview.window4; title=none
Success

show multiviews titles config
multiview(mv1);
  multiview.gen; audioSourceWindow=none, canvasWidth=3840,
canvasHeight=2160
  multiview.window1; position=bottomCenter, textSize=8,
textColor=lightGray, backgroundColor=black, textTransparency=0,
backgroundTransparency=80
  multiview.window2; position=bottomCenter, textSize=8,
textColor=lightGray, backgroundColor=black, textTransparency=0,
backgroundTransparency=80
  multiview.window3; position=bottomCenter, textSize=8,
textColor=lightGray, backgroundColor=black, textTransparency=0,
backgroundTransparency=80
  multiview.window4; position=bottomCenter, textSize=8,
textColor=lightGray, backgroundColor=black, textTransparency=0,
backgroundTransparency=80
Success
```

show preset

Shows information about a preset

Syntax

```
show preset name arg since
```

Parameters

name

Type: STRING

The name of the preset

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`commandBlob`	Displays command list in "blob" format. Commands separated by semi-colons.
`commands`	Shows current list of preset commands
`config`	Shows preset description
`runLog`	Shows information about last time preset was run
`schedule`	Shows schedule details for the preset
`status`	Displays text configuration information for multiview.

Examples

```
show preset test1 schedule all
preset(test1);
  preset.schedule.today; mode=enabled, color=#652d90, month=all,
dayOfMonth=all, dayOfWeek=weekday, hour=14, minute=30
lastChangeIdMax(92);
Success

show preset test1 runLog since 0
preset(test1);
lastChangeIdMax(92);
Success
```

Examples

```
show preset test1 config since 0
preset(test1);
  preset.gen; description=Playing with preset
lastDeleteIdMax(3);
lastChangeIdMax(90);
Success

show preset test1 commands since 0
preset(test1);
  preset.line1; cmd=join MediaPlayer Bot_Right fastSwitched
  preset.line2; cmd=join none Bot_Right analogAudio
  preset.line3; cmd=join videoSource Bot_Right hdmiAudio
  preset.line4; cmd=set decoder Bot_Right hdmiAudioOut source
hdmiAudio
  preset.line5; cmd=join mv2x2-Art Top-Right multiview
  preset.line6; cmd=join none Top-Right analogAudio
  preset.line7; cmd=join videoSource Top-Right hdmiAudio
  preset.line8; cmd=join mv3x3-Art Top_Left multiview
  preset.line9; cmd=join none Top_Left analogAudio
  preset.line10; cmd=join videoSource Top_Left hdmiAudio
  preset.line11; cmd=join mv4x4-Art Bot-Left multiview
  preset.line12; cmd=join none Bot-Left analogAudio
  preset.line13; cmd=join videoSource Bot-Left hdmiAudio
lastChangeIdMax(94);
Success

show preset test1 commandBlob since 0
preset(test1);
  preset.cmdBlob; cmdBlob=join MediaPlayer Bot_Right
fastSwitched;join none Bot_Right analogAudio;join videoSource
Bot_Right hdmiAudio;set decoder Bot_Right hdmiAudioOut source
hdmiAudio;join mv2x2-Art Top-Right multiview;join none Top-Right
analogAudio;join videoSource Top-Right hdmiAudio;join mv3x3-Art
Top_Left multiview;join none Top_Left analogAudio;join videoSource
Top_Left hdmiAudio;join mv4x4-Art Bot-Left multiview;join none Bot-
Left analogAudio;join videoSource Bot-Left hdmiAudio
lastChangeIdMax(94);
Success
```

Related Commands

create preset
delete preset
run preset
set preset

show responses

Displays response strings from the specified device.

Syntax

`show responses` *`id type param3`*

Parameters

`id`

Type: STRING or MAC Address

The name or MAC address of the device. String names are case-sensitive.

`type`

Type: STRING

Supply one of the following arguments.

argument	Description
`ir`	Displays IR response strings.
`rs232`	Displays RS232 response strings.

`param3`

Supply one of the following arguments.

argument	Description
`last`	Displays the last received response, based on the argument supplied for the *`type`* parameter.
`lastChangeId`	Displays the **`lastChangeId`** of the most recently received response.
`since`	Displays only new response data. Follow this argument with desired value to query.

Example

```
show responses 0:1e:c0:f6:b0:8a rs232 since 10
lastChangeId(0);
Success

show responses 0:1e:c0:f6:b0:8a ir lastChangeId
lastChangeId(0);
Success

show responses 0:1e:c0:f6:b0:8a ir last
lastChangeId(0);
Success

show responses UHDdec1 rs232 last
device(34:1b:22:80:64:68);
  device.rs232Response.19; string=”Yes ZeeVee Support is the
Greatest\x0D”
lastChangeId(20);
Success

Zyper$ show responses UHDdec1 rs232 since 19
device(34:1b:22:80:64:68);
  device.rs232Response.19; string=”Yes ZeeVee Support is the
Greatest\x0D”
  device.rs232Response.20; string=”Really, still the greatest!\x0D”
lastChangeId(21);
Success
```

show role

Shows information about a specific role or all roles.

Syntax

`show role rolename|all maxAccess[since]`

Parameters

role

Type: STRING

The name of the role

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
since	Required when using this parameter.

Examples

```
show role admin maxAccess since 0
role(admin);
  role.account; maxAccess=admin
  role.device; maxAccess=admin
  role.log; maxAccess=admin
  role.multiview; maxAccess=admin
  role.netmap; maxAccess=admin
  role.preset; maxAccess=admin
  role.role; maxAccess=admin
  role.server; maxAccess=admin
  role.snmpagent; maxAccess=admin
  role.tls; maxAccess=admin
  role.videowall; maxAccess=admin
  role.zone; maxAccess=admin
lastChangeIdMax(12);
Success
```

```
-----------------------------------------------------------

show role all maxAccess
role(admin);
  role.account; maxAccess=admin
  role.device; maxAccess=admin
  role.log; maxAccess=admin
  role.multiview; maxAccess=admin
  role.netmap; maxAccess=none
  role.preset; maxAccess=admin
  role.role; maxAccess=admin
  role.server; maxAccess=admin
  role.snmpagent; maxAccess=admin
  role.tls; maxAccess=admin
  role.videowall; maxAccess=admin
  role.zone; maxAccess=admin
role(Jr.Tech);
  role.account; maxAccess=view
  role.device; maxAccess=join
  role.log; maxAccess=view
  role.multiview; maxAccess=none
  role.netmap; maxAccess=none
  role.preset; maxAccess=join
  role.role; maxAccess=none
  role.server; maxAccess=none
  role.snmpagent; maxAccess=none
  role.tls; maxAccess=none
  role.videowall; maxAccess=join
  role.zone; maxAccess=view
lastChangeIdMax(2281);
Success
```

Related Commands

```
create role
delete role
set role rolename
set account username role
```

show server config

Displays configuration information for the Management Platform.

Syntax

```
show server config [since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show server config
server(192.168.0.21);
  server.gen; autoEdidMode=enabled, redundancy=enabled
  server.ipServerAddress; mode=static, address=192.168.0.21,
mask=255.255.255.0, gateway=0.0.0.0, dns=0.0.0.0
  server.ipManagementAddress; mode=static, address=192.168.20.2,
mask=255.255.255.0, gateway=0.0.0.0, dns=0.0.0.0
  server.ntpServer; address=pool.ntp.org
  server.telnetAccess; mode=enabled
  server.ftpAccess; mode=enabled
  server.encoderDefault.edid; audio=onlyPcm
  server.dataTunnelMode; telnet=telnetHandshakeMode
  server.logging; level=1
  server.isaac; address=none, subsystemId=none
  server.discover; mode=broadcast
lastChangeIdMax(543);
Success
```

Related Commands

`show server info`

show server info

Displays information for the Management Platform, including IP settings, uptime, and license level.

Syntax

```
show server info [since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show server info
server(192.168.0.21);
  server.gen; hostname=zyper.local, serverType=NUC-E,
 version=3.3.40029, previousVersion=3.2.39797, master=true,
 location=Wiring closet, contact=ZeeVee
  server.gen; uptime=3d:0h:45m:32s, freeMem= 2.60GB,
sdvoeVersion=3.6.0.1, bootCount=19, serialNumber=ZZM1MC00001A
  server.gen; macAddress=00:0e:c6:8e:38:6f,
managementMacAddress=54:b2:03:f0:ee:99
  server.ipActive; ipServerAddr=192.168.0.21,
ipManagementAddr=0.0.0.0, gatewayAddr=0.0.0.0, dnsAddr=0.0.0.0
  server.ipActive; managementGatewayAddr=0.0.0.0,
managementDnsAddr=0.0.0.0
  server.time; time="Mon Apr  8 09:38:06 2024", timezone=America/
New_York
  server.license; productID=367d8850-f6f2-cd73-fbb1-54b203f0ee99,
 license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=8, allDevices=14,
allDevicesUp=9, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_1.7.24.0=4, num_1.7.31.0=1,
num_2.0.0.0=2, num_2.1.0.0=2
lastChangeIdMax(535);
Success
```

Related Commands

`show server config`
`revert server`

show server ip duplicates

Shows if there any duplicate IP addresses in the system. Can include encoders, decoders, ICRON or Dante units

Syntax

```
show server ip duplicates [since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show server ip duplicates
server(192.168.0.22);
lastChangeIdMax(88);
Success
```

Related Commands

`show server config`

show server managedDevices

Shows a listing of all devices currently under managment by this Management Server. Use this command with set server isolationMode and set server discoverMode.

Syntax

```
show server managedDevices all[since]
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show server managedDevices all
Success
```

Related Commands

```
show server ip duplicates
set server isolationMode
set server discoverMode
```

show server redundancy

Displays information about master and slave Management Platforms

Syntax

```
show server redundancy
```

Parameters

since

This parameter is optional and can be specified to display units based on the number of changes, using the `lastChangeId` value on each device. However, if used, a lastChangeId value must follow. Supply the `since` argument before the providing the `lastChangeId` value.

argument	Description
`since`	Required when using this parameter.

Example

```
show server redundancy
server(172.16.6.111);
  server.status; state=master, version=2.1.1.36527, wasMaster=true,
wasSlave=true
  server.config; preferredMaster=true, preferredSlave=true
  server.virtualIp; address=0.0.0.0, networkInterface=video
Success
```

Related Commands

`set server redundancy`
`redundancy switchover`

show snmp

Displays information related to SNMP. (Please see Section 5 of this manual for additional details on SNMP support)

Syntax

```
show snmp arg
```

Parameters

type

Type: STRING

Supply one of the following arguments.

argument	Description
`trapServers`	Displays snmp trap servers.
`users`	Displays snmp users.

Example

```
show snmp trapServers
snmp(172.16.6.111);
Success

show snmp users
snmp(172.16.6.111);
Success
```

Related Commands

add snmp
delete snmp

show snmp netNode

Displays information related to SNMP NetNodes use with the ZeeVee Visualization, Analysis and Monitoring (VAM) tool.

Syntax

```
show snmp ident arg
```

Parameters

ident

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays info for all netNodes.
`Name`	Name of specific netNode

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`general`	Basic information
`snooping`	Snooping and Querier information
`warnings`	Warnings associated with setup/ configuration
`justChanges since <lastChangeID : lastChangeNumber>`	Changes to netNodes sinces a given changeID. (Command only valid if "ident" = all)

Examples

```
show snmp netNode all snooping
netNode(1);
  netNode.snooping; mode=NA, proxyQuerier=NA
netNode(4);
  netNode.snooping; mode=enabled, proxyQuerier=disabled
Success
```

```
show snmp netNode all general
netNode(1);
  netNode.gen; name=”myZMPserver”, chassisId=54:b2:03:f0:ee:99,
type=NA, portErrors=0, ip=192.168.0.21, state=up,
uptime=0d:1h:27m:12s
  netNode.gen; make=zeevee, descr=”ZyPer Management Platform”,
version=” 3.3.39589”, location=”Wiring closet”
netNode(4);
  netNode.gen; name=”Arts_8x8”, chassisId=8c:3b:ad:68:6b:34,
type=bridge+router, portErrors=1, ip=192.168.0.1, state=up,
uptime=3d:6h:48m:38s
  netNode.gen; make=netgear, descr=”M4300-8X8F ProSAFE 8-port
10GBASE-T and 8-port 10G SFP+”, version=” 12.0.17.13”,
location=”Dining Room”
Success

show snmp netNode all warnings
netNode(1);
netNode(4);
  netNode.warning.1; message=Port-level proxy querier is enabled;
it will cause issues in a multi-switch network
  netNode.warning.2; message=VLAN-level proxy querier is enabled;
this will cause capacity issues on inter-switch trunks
Success
```

Related Commands

add snmp netNode
delete snmp netNode

show snmp netNode vlan

Displays information related to SNMP NetNodes use with the ZeeVee Visualization, Analysis and Monitoring (VAM) tool.

Syntax

```
show snmp ident vlan arg snooping
```

Parameters

ident

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays info for all netNodes.
`Name`	Name of specific netNode

arg

Type: STRING or INTEGER

Supply one of the following arguments.

argument	Description
`VLAN Number`	Number of the VLAN. Example "2"
`all`	all VLANs

Example

```
show snmp netNode all vlan all snooping
netNode(1);
netNode(4);
  netNode.vlan.1; snooping=enabled, agingSeconds=600,
fastLeave=enabled, proxyQuerier=enabled
Success
```

Related Commands

add snmp netNode
delete snmp netNode

show snmp netNode port

Displays information related to SNMP NetNodes use with the ZeeVee Visualization, Analysis and Monitoring (VAM) tool.

Syntax

`show snmp` *`ident`* `port` *`arg`* *`detail`*

Parameters

`ident`

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays info for all netNodes.
`Name`	Name of specific netNode

`arg`

Type: STRING or INTEGER

Supply one of the following arguments.

argument	Description
`Port Number`	Number of the Port. Example "2"
`all`	all Ports

`arg`

Type: STRING or INTEGER

Supply one of the following arguments.

argument	Description
`state`	Displays status of given port
`peer`	Detailed information on given port including peerSlot and peerPort
`snooping`	Snooping status including fastLeave, Querier, mrouter, aging and flooding of unregistered multicasts
`vlan`	Shows VLAN IDs
`stats`	Shows MTU, link speeds, data in/out rates

Example

```
show snmp netNode all port all state
netNode(1);
  netNode.port.state;++       state bandwidth
  netNode.port.state;;;       ----- ---------
  netNode.port.state.0.2;     down  ok
  netNode.port.state.0.1;     up    ok
  netNode.port.state;--
netNode(4);
  netNode.port.state;++       state bandwidth
  netNode.port.state;;;       ----- ---------
  netNode.port.state.0.1;     down  ok
  netNode.port.state.0.2;     down  ok
  netNode.port.state.0.3;     down  ok
  netNode.port.state.0.4;     down  ok
  netNode.port.state.0.5;     down  ok
  netNode.port.state.0.6;     down  ok
  netNode.port.state.0.7;     down  ok
  netNode.port.state.0.8;     down  ok
  netNode.port.state.0.9;     up    ok
  netNode.port.state.0.10;    up    ok
  netNode.port.state.0.11;    down  ok
  netNode.port.state.0.12;    down  ok
  netNode.port.state.0.13;    up    ok
  netNode.port.state.0.14;    up    ok
  netNode.port.state.0.15;    up    ok
  netNode.port.state.0.16;    up    ok
  netNode.port.state;--
Success
```

Related Commands

add snmp netNode
delete snmp netNode

show snmp netNode multicastForwardingDb

Displays information related to SNMP NetNodes use with the ZeeVee Visualization, Analysis and Monitoring (VAM) tool.

Syntax

```
show snmp ident multicastForwardingDb
```

Parameters

ident

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays info for all netNodes.
`Name`	Name of specific netNode

Related Commands

`add snmp netNode`
`delete snmp netNode`

Example

```
show snmp netNode all multicastForwardingDb
netNode(1);
netNode(4);
  netNode.mfdb;++   group              encoder              stream      portsOut
  netNode.mfdb;;;   ------------------ -------------------- ----------- ----------------
----
  netNode.mfdb.1;   1:0:5e:0:0:12      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.2;   1:0:5e:0:0:e6      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.3;   1:0:5e:0:0:e7      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.4;   1:0:5e:0:0:e9      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.5;   1:0:5e:0:0:fb      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.6;   1:0:5e:0:1:81      unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.7;   1:0:5e:0:1:bc      unknown              unknown     [0.9]
  netNode.mfdb.8;   1:0:5e:11:13:bd    unknown              unknown     [0.10]
  netNode.mfdb.9;   1:0:5e:11:13:c0    unknown              unknown     [0.10]
  netNode.mfdb.10;  1:0:5e:11:1c:84    unknown              unknown     [0.10]
  netNode.mfdb.11;  1:0:5e:1:0:0       unknown              unknown     [0.10]
  netNode.mfdb.12;  1:0:5e:1:0:1       unknown              unknown     [0.10]
  netNode.mfdb.13;  1:0:5e:1:1:6       XSE_Enc_Combo        hdmiAudio   [0.13]
  netNode.mfdb.14;  1:0:5e:1:1:7       XSE_Enc_Combo        danteAudio  [0.13]
  netNode.mfdb.15;  1:0:5e:1:1:fd      unknown              unknown     [0.14-0.15]
  netNode.mfdb.16;  1:0:5e:1:1:fe      unknown              unknown     [0.13,0.16]
  netNode.mfdb.17;  1:0:5e:21:13:bd    UHD60-2EA            video       [0.10]
  netNode.mfdb.18;  1:0:5e:21:13:c0    UHD60-1EA            video       [0.10]
  netNode.mfdb.19;  1:0:5e:21:1c:84    UHD60-Wallplate      video       [0.10]
  netNode.mfdb.20;  1:0:5e:21:21:21    unknown              unknown     [0.9]
  netNode.mfdb.21;  1:0:5e:2:0:0       unknown              unknown     [0.9]
  netNode.mfdb.22;  1:0:5e:31:13:bd    UHD60-2EA            hdmiAudio   [0.10]
  netNode.mfdb.23;  1:0:5e:31:13:c0    UHD60-1EA            hdmiAudio   [0.10]
  netNode.mfdb.24;  1:0:5e:31:1c:84    UHD60-Wallplate      hdmiAudio   [0.10]
  netNode.mfdb.25;  1:0:5e:41:13:bd    unknown              unknown     [0.10]
  netNode.mfdb.26;  1:0:5e:41:13:c0    unknown              unknown     [0.10]
  netNode.mfdb.27;  1:0:5e:41:1c:84    unknown              unknown     [0.10]
  netNode.mfdb.28;  1:0:5e:41:d1:bc    unknown              unknown     [0.10]
  netNode.mfdb.29;  1:0:5e:51:13:bd    unknown              unknown     [0.10]
  netNode.mfdb.30;  1:0:5e:51:13:c0    unknown              unknown     [0.10]
  netNode.mfdb.31;  1:0:5e:51:1c:84    unknown              unknown     [0.10]
  netNode.mfdb.32;  1:0:5e:61:f9:88    unknown              unknown     [0.10]
  netNode.mfdb.33;  1:0:5e:71:13:bd    unknown              unknown     [0.10]
  netNode.mfdb.34;  1:0:5e:71:13:c0    unknown              unknown     [0.10]
  netNode.mfdb.35;  1:0:5e:71:1c:84    unknown              unknown     [0.10]
  netNode.mfdb.36;  1:0:5e:7f:2d:d8    unknown              unknown     [0.10,0.15]
  netNode.mfdb.37;  1:0:5e:7f:ff:fa    unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.mfdb.38;  1:0:5e:7f:ff:ff    unknown              unknown     [0.10,0.13,0.15]
  netNode.mfdb.39;  91:e0:f0:1:0:0     unknown              unknown     [0.9-0.10,0.13-
0.16]
  netNode.port.mfdb;--
Success
```

show tls ca pem

Shows Transport Layer Security information for the Certificate Authority

Syntax

```
show tls ca pem arg
```

Parameters

type

Type: STRING

Supply one of the following arguments.

argument	Description
cert	Displays the certificate
privKey	Displays the RSA Private Key
signedCert	Display the signed certificate

Examples

```
show tls ca pem cert
pemData:
-----BEGIN CERTIFICATE-----
..................
-----END CERTIFICATE-----
Success

show tls ca pem privKey
pemData:
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,10E7EB7C47A3B07D64608BC1D4A63F5F
................
-----END RSA PRIVATE KEY-----
Success

show tls ca pem signedCert
pemData:
-----BEGIN CERTIFICATE-----
..................
-----END CERTIFICATE-----
Success
```

Related Commands

show tls summary

show tls device pem

Shows Transport Layer Security information for the Server or Device

Syntax

```
show tls device pem target arg
```

Parameters

target

Type: STRING

Supply one of the following arguments.

argument	Description
device	show Device information
radius	show Radius information
server	show Server information

arg

Type: STRING

Supply one of the following arguments.

argument	Description
csr	Displays certificate signing request
cert	Displays the certificate (Device and Server only)
privKey	Displays RSA private key
caIntermediates	Displays chain of certificates between root cert and your cert. Used when cert received from trusted certificate authority such as Verisign. (Device and Server only)

Example

```
show tls device pem server csr
pemData:
-----BEGIN CERTIFICATE REQUEST-----
.............
-----END CERTIFICATE REQUEST-----
Success
```

show tls summary

Shows a summary of Transport Layer Security settings.

Syntax

```
show tls summary
```

Parameters

none or

argument	Description
ca	show Certificate Authority information
radius	show Radius information
server	show Server information

Examples

```
show tls ca summary
server(192.168.0.21);
  ca.tls.caCert; status=valid, C=US, CN=caCert, L=Billerica,
 O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  ca.tls.caCert; issuer=caCert
  ca.tls.caCert; fingerprint=8A783DC9351E8877BBE5D0C1E7A913336B15
0F68
  ca.tls.caCert; expires=01/09/34T13:55:55-0500
  ca.tls.signed; status=valid, C=US, CN=zeevee.com, O=ZeeVee,
 OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  ca.tls.signed; issuer=caCert
6Dca.tls.signed; fingerprint=44CD6AB55BAB13AF3B7A76E453A307761FDBCB
  ca.tls.signed; expires=01/11/25T13:58:19-0500
Success


show tls radius summary
server(192.168.0.21);
  radius.tls.csr; status=valid, C=US, CN=zeevee.com, O=ZeeVee,
 OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
Success
```

```
show tls summary
server(192.168.0.21);
  server.tls.server; tlsMode=disabled, fqdnMode=fromCert, fqdn=NA
  server.tls.csr; status=invalid
  server.tls.serverCert; status=invalid
  server.tls.caChainCert; status=invalid
  ca.tls.caCert; status=valid, C=US, CN=caCert, L=Billerica,
 O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  ca.tls.caCert; issuer=caCert  ca.tls.caCert; fingerprint=8A783DC93
51E8877BBE5D0C1E7A913336B150F68
  ca.tls.caCert; expires=01/09/34T13:55:55-0500
  ca.tls.signed; status=valid, C=US, CN=zeevee.com, O=ZeeVee,
 OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  ca.tls.signed; issuer=caCert
6Dca.tls.signed; fingerprint=44CD6AB55BAB13AF3B7A76E453A307761FDBCB
  ca.tls.signed; expires=01/11/25T13:58:19-0500
  device.tls.csr; status=valid, C=US, CN=zeevee.com, O=ZeeVee,
 OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  device.tls.deviceCert; status=valid, C=US, CN=zeevee.com,
 O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  device.tls.deviceCert; issuer=caCert
  device.tls.deviceCert; fingerprint=44CD6AB55BAB13AF3B7A76E453A3077
61FDBCB6D
  device.tls.deviceCert; expires=01/11/25T13:58:19-0500
  device.tls.deviceCaChainCert; status=valid, C=US, CN=zeevee.com,
 O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
  device.tls.deviceCaChainCert; issuer=caCert
  device.tls.deviceCaChainCert; fingerprint=44CD6AB55BAB13AF3B7A76E4
53A307761FDBCB6D
  device.tls.deviceCaChainCert; expires=01/11/25T13:58:19-0500
  radius.tls.csr; status=valid, C=US, CN=zeevee.com, O=ZeeVee,
 OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
Success
```

Related Commands

```
show tls server pem privKey
show tls ca pem privKey
load tls ca cert
load tls ca privateKey
load tls server
generate tls ca privKeyPass
generate tls server csr privKeyPass
```

show tls device summary

Shows a summary of Transport Layer Security settings for Device

Syntax

```
show tls device summary
```

Parameters

none

Examples

```
show tls device summary
server(192.168.0.21);
  device.tls.csr; status=valid, C=US, CN=zeevee.com,
 O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com
device.tls.deviceCert; status=invalid
  device.tls.deviceCaChainCert; status=valid, C=US, CN=caCert,
 L=Billerica, O=ZeeVee, OU=Money, ST=MA,
emailAddress=aweeks@zeevee.com
  device.tls.deviceCaChainCert; issuer=caCert
  device.tls.deviceCaChainCert;
fingerprint=A7C444401F258372BEB6E015EE1FC4E357AB4E89
  device.tls.deviceCaChainCert; expires=01/06/34T15:24:05-0500
  device.tls.deviceCaCert; status=invalid
Success
```

Related Commands

```
show tls server pem privKey
show tls ca pem privKey
load tls ca cert
load tls ca privateKey
load tls server
generate tls ca privKeyPass
generate tls server csr privKeyPass
```

show values

Shows all possible information/values associated with encoders, decoders, servers or multiviews.

Syntax

```
show values arg
```

Parameters

arg

Type: STRING

Supply one of the following arguments.

argument	Description
`all`	Displays all values encoders, decoders, servers and multiviews. (Status, Config, Info, Redundancy)
`encoder status`	Shows values associated with encoder status
`encoder config`	Shows values associated with encoder config
`decoder status`	Shows values associated with decoder status
`decoder config`	Shows values associated with decoder config
`server info`	Shows values associated with server info
`server config`	Shows values associated with server config
`server redundancy`	Shows values associated with server redundancy
`multiview status`	Shows values associated with multiview status
`multiview config`	Shows values associated with multiview config

Examples

```
show values server config
values(serverConfig);
  server.ipServerAddress.mode; values=dhcp|static
  server.ipManagementAddress.mode; values=none|dhcp|static
  server.telnetAccess; values=enabled|disabled
  server.encoderDefault.edid.audio; values=onlyPcm|allowCompressed
  server.dataTunnelMode; values=telnet|raw
  server.logging; values=<integer 1-5>
Success
```

Examples

```
show values encoder config
values(encoderConfig);
  device.gen.ethernetManagementPortMode; values=enabled|disabled
  device.gen.controlAuthenticationMode; values=enabled|disabled
  device.gen.name; values=<string 1-256>
  device.ip.mode; values=dhcp|static|linkLocal
  device.ip.address; values=<IPv4Address>
  device.ip.mask; values=<IPv4Mask>
  device.ip.gateway; values=<IPv4Address>|NA
  device.rs232.baud; values=2400|9600|19200|38400|57600|115200
  device.rs232.parity; values=none|even|odd
  device.ports.videoPort; values=auto|initializing|unknown|hd
mi|displayPort|hdmiOptionalIn|usbc|vga|component|composite|s-
video|analogNone|hdsdi|12gsdi
  device.analogAudioStream.mode; values=enabled|disabeld
  device.audioOutSourceType.analogOutSourceType; values=analogAudio
|hdmiAudioDownmix
  device.edid.loadMode; values=auto|file
  device.edid.audio; values=onlyPcm|allowCompressed|serverDefault
  device.hdmi.hdcpMode; values=enabled|disabled|enabled1.4
  device.hdmiAudioStream.mode; values=enabled|disabeld
  device.previewStream.mode; values=enabled|disabeld
  device.previewStream.type; values=hls|jpeg
  device.previewStream.width; values=auto|<integer 180-400>
  device.usb.filter; values=none|exceptHid|storage
  device.videoStream.mode; values=enabled|disabeld
  device.videoScaledStream.mode; values=enabled|disabeld
Success
```

show videoWalls

Displays all video walls that have been created and all associated information.

Syntax

```
show videoWalls
```

Parameters

```
none
```

Example

```
show videoWalls
videoWall(wall1);
  videoWall.gen; videoSourceMac=none, numDisplayRows=2,
numDisplayCols=2
  videoWall.bezel; top=0, bottom=0, left=0, right=0
  videoWall.decodersRow1; col1=Top_Left, col2=Top-Right
  videoWall.decodersRow2; col1=Bot-Left, col2=Bot_Right
Success
```

Related Commands

```
create videoWall
set videoWall size
```

show zones

Displays all zones that have been created and all associated information.

Syntax

```
show zones
```

Parameters

none

Example

```
show zones
  1stfloor; Top-Right, Top_Left
  1stfloor.1stfloorroom2; empty
Success
```

Related Commands

add zoneDisplay
create zone
delete zone
delete zoneDisplay

shutdown server

Performs a shutdown of the Management Platform.

Syntax

```
shutdown server
```

Parameters

none

Example

```
shutdown server
Success
Connection closed by foreign host.
```

---

sign tls csr caPrivateKeyPass

Use the CSR to create signed TLS certificate

Syntax

```
sign tls csr caPrivateKeyPass * fromInput *

sign tls csr PrivateKeyPass * fromFile filename
```

Parameters

input

Type: STRING

String representing the Private Key Password. The system will promt for a string input. This should be the PEM data.

filename

Type: STRING

The name of the PEM data file to load. (Must already exist on ZMP in Files directory)

Example

```
sign tls csr caPrivateKeyPass * fromInput *
Enter passphrase: ******
Enter PEM text (ctr-d to end):
-----BEGIN CERTIFICATE REQUEST-----
............
-----END CERTIFICATE REQUEST-----
Success
```

Notes:

File must be previously copied onto ZMP into the Files directory using FTP.

Related Commands

show cls summary
show tls pem ca signedCert

sleep

Specifies a sleep duration in milliseconds. This command is sometime required when executing a series of commands within a web page, using AJAX. Often times, a pause must occur in order for a device or the Management Platform to change states before another command is executed.

Syntax

`sleep` *`ms`*

Parameters

`ms`

Type: INTEGER

The duration in milliseconds.

Example

```
sleep 500
Success
```

Related Commands

`script`

start encoder

Used to start a specific encoder multicast stream. This command only has affect if at least one decoder has been "joined" to the encoder and the "encoder stop" command has been used to override the enabling of the encoder stream. In effect, this command removes a previously entered "encoder stop" command – it returns stream control to normal operation based on existing "join" configuration. The command will immediately restore stream operation based on existing join configuration. No further join commands are required. (ZyPer4K family only)

Syntax

```
start encoder id stream arg
```

Parameters

id

Type: STRING or MAC Address

The identifier of the device. Either the full or portion of a string name or MAC address can be supplied.

arg

Supply one of the following arguments.

argument	Description
`analogAudio`	analog audio multicast stream.
`hdmiAudio`	downmix audio multicast stream
`video`	full scale video stream
`videoScaled`	downscaled video stream (for multiview)

Example

```
start encoder Myencoder1 stream video
Success
```

Related Commands

`stop encoder`

start kvm

Used to start a specific kvm. (ZyPer4K XS/XSE/XR only)

Syntax

```
start kvm name
```

Parameters

name

Type: STRING

The name of the kvm. String names are case-sensitive.

Example

```
start kvm ControlBoard
Success
```

Related Commands

```
create kvm
delete kvm
set kvm
show kvms
stop kvm
```

stop encoder

Used to stop a specific encoder multicast stream. This command only has affect if at least one decoder has been "joined" to the encoder. In effect, this command overrides any existing "join" command – either present or future. (ZyPer4K family only)

When stopping a "scaled-video" stream, any multiview window receiving that stream will go black. The rest of the multiview will be unaffected.

Syntax

`stop encoder` *`id`* `stream` *`arg`*

Parameters

`id`

Type: STRING or MAC Address

The identifier of the device. Either the full or portion of a string name or MAC address can be supplied.

`arg`

Supply one of the following arguments.

argument	Description
`analogAudio`	analog audio multicast stream.
`hdmiAudio`	downmix audio multicast stream
`video`	full scale video stream
`videoScaled`	downscaled video stream (for multiview)

Example

```
stop encoder Myencoder1 stream videoScaled
Success
```

Related Commands

`start encoder`

stop kvm

Used to stop a specific kvm. (ZyPer4K XS/XSE/XR only)

Syntax

```
stop kvm name
```

Parameters

name

Type: STRING

The name of the kvm. String names are case-sensitive.

Example

```
stop kvm ControlBoard
Success
```

Related Commands

```
create kvm
delete kvm
set kvm
show kvms
stop kvm
```

switch

This command is used in conjunction with the IR and RS232 switching commands. Both the `rs232` and the `ir` argument specify unidirectional connection between two devices. When switching data to the server, use the `show responses` command to retrieve the data.

Syntax

`switch txid rxid type`

Parameters

`txid`

Type: STRING or MAC Address

The name or MAC address of the encoder. String names are case-sensitive.

`rxid`

Type: STRING or MAC Address

The name or MAC address of the decoder. String names are case-sensitive.

`type`

Type: STRING

Supply one of the following arguments.

argument	Description
`ir`	Specifies a IR connection. (ZyPer4K family and ZyPerUHD only)
`rs232`	Connection to another device or the server. Set *`rxid`* = `none` to pass data to an arbitrary IP host.

Example

```
switch Wildlife SonyXBR4 rs232
Success
```

Related Commands

`send`

troubleReport

Generates capture logs and system state information and is used by the ZeeVee support team for troubleshooting purposes. This unencrypted file is in `.tgz` format and is written to the `/srv/ftp/files` folder on the Management Platform.

If using password option; the encrypted file is in `.gpg` format and written to the same location.

Syntax

```
troubleReport
troubleReport password pw
```

Parameters

pw

Type: STRING

Password to open the encrypted Trouble Report file.

Note the password is optional feature and will create an encrypted trouble report file.

Examples

```
troubleReport password 1234
Clean up files
Creating Trouble Report
Saving device status and configuration...
Saving SQL database...
Saving system files...
Saving device EDIDs...
Saving device specific information; this may take a few seconds...
Success

troubleReport
Clean up files
Creating Trouble Report
Saving device status and configuration...
Saving SQL database...
Saving system files...
Saving device EDIDs...
Saving device specific information; this may take a few seconds...
Success
```

update device

Updates the firmware on the encoder and/or decoder units. The firmware update file uses the `.apz or .zip` extension.

Syntax

```
update device arg file
```

Parameters

arg

Supply one of the following arguments.

argument	Description
`id`	Encoder or Decoder name. Names are case-sensitive
`all`	All encoders and decoders in the system
`encoders`	All encoders in the system
`decoders`	All decoders in the system

file

Type: STRING

The full filename of the software file.

Example

```
update device all Z4K_Firmware_HDMI2.0_v4_1_2_9.apz
Warning:(18) Firmware updating started, use 'show device status' to
monitor progress
Success
```

update server

Updates the Management Platform software. The server software file uses the `.zyper` extension. Refer to Updating the Software (page 412) for more information on using this command.

Syntax

```
update server file
```

Parameters

file

Type: STRING

The full filename of the software file.

Example

```
update server new-sofware-file.zyper
Success

Server rebooting; connection will end
```

Important Note:

The ZyPer MP update file will be available in five, platform-specific versions. Please use the correct version for the hardware platform being updated.

ZyPerMP Single NIC NUC computer: update_nuc_4.0.xxxxx.zyper
ZyPerMP Dual NIC NUC computer: update_nuc2004_4.1.xxxxx.zyper
ZyPerMP Proserver: update_proserver_4.1.xxxxx.zyper
ZyPerMP VMware: update_vm_4.1.xxxxx.zyper
ZyPerUHD60-EMP: update_zemp_4.1.xxxxx.zyper

4 Event Mechanism

events

ZMP Event Mechanism

There are three ways to receive events:

- Second telnet session to receive events asynchronously. Session not used for API commands, only to receive events.
- Browser WebSocket to ZMP server. Allows server to asynchronously send events to the browser.
- Reliable, low-overhead API command to poll for events.

Event Message Format

```
Event::<name>::<source>::<date>::<lastChangeId>::: <Message>
```

Where:

source device-name or "server"

Example:

Event::DeviceStateChange::DE1(d8:80:39:9a:af:e1)::Jun-18-02:42:56:PM::13:::
state=Up

Telnet Event Session

- Client telnets to the ZMP server as normal
- API prompt received
- Command entered: "events"
  - o Causes the event mode to be entered
  - o Server sends initial events (described below) and new events as they occur to this telnet session
  - o Any character entered to the server causes the mode to exit back to the API prompt

Browser WebSocket

Client usage of a WebSocket to receive events is quite simple. Example JavaScript from the sample zyper.html file shows how to connect to the websocket server on ZMP. Upon connection, initial events (described below) will be sent, and then any new events as they occur.

```
eventSock = new WebSocket("ws://rey:8001", "zeeVeeLogging"
eventSock.onopen = eventSockOpened;
eventSock.onMessage = eventRcvd;
eventSock.onclose = eventSockClosed;

function eventSockOpened() {
        eventSock.send("Send Events");   // igonored by server
}
function eventRcvd(event) {
        var evWin = document.getElementById('eventWindow');
        evWin.innerHTML += event.data + "<br>";
        evWin.scrollTop = evWin.scrollHeight;
}
function eventSockClosed() {
        console.log("EVENT SOCK CLOSED");
}
```

API Polling

The "show events since <id>" command may be used over telnet or from a browser using AJAX/JSON and preferably long-polling. It is a simple, low overhead and very reliable mechanism to ensure all events have been received.

```
Zyper$ show events since 0
server(172.16.2.169);
    server.event.0; event="Mon Jun 18 18:44:06 2018: ip=172.16.2.64, state=down-
    >master"
    server.event.1; event="Mon Jun 18 18:44:10 2018: state=up"
    server.event.2; event="Mon Jun 18 18:44:10 2018: state=up"
    ...
    server.event.28; event="Tue Jun 19 05:00:03 2018: sizeX=1280->720,
    sizeY=720->480, fps=60.00->60.00"
    server.event.29; event="Tue Jun 19 05:01:24 2018: cable=disconnected"
    server.event.30; event="Tue Jun 19 05:01:26 2018: cable=connected"
    server.event.31; event="Tue Jun 19 05:01:26 2018: sizeX=720->1280,
    sizeY=480->720, fps=60.00->60.00"
lastChangeId(32);
Success
Zyper$
Zyper$ show events since 28
server(172.16.2.169);
    server.event.28; event="Tue Jun 19 05:00:03 2018: sizeX=1280->720,
    sizeY=720->480, fps=60.00->60.00"
    server.event.29; event="Tue Jun 19 05:01:24 2018: cable=disconnected"
    server.event.30; event="Tue Jun 19 05:01:26 2018: cable=connected"
    server.event.31; event="Tue Jun 19 05:01:26 2018: sizeX=720->1280,
    sizeY=480->720, fps=60.00->60.00"
lastChangeId(32);
Success
Zyper$
Zyper$ show events since 32
lastChangeId(32);
Success
Zyper$
```

Initial Events

Upon entering telnet "events" mode, or upon a WebSocket connection, the server will send a DeviceStatus event for each known device. Each of these events will have lastChangeId set to 0.

Event List

ServerIpChanged:

Message: `ipAddress=<from>-><to>`

Example:

Event::ServerIpChanged::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
ipAddress=169.254.1.10->172.16.2.22

ServerStateChange:

Message: `state=<from>-><to>`

serverState:

down
initialization
master
slave-sync
slave
slave-switching-over
slave-db-updating
not-participating
slave-waiting-for-master
slave-version-mismatch

Example:

Event::stateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=slave->master

NewServer:

Message: `id=<id>, ip=<ipAddr>, state=<serverState>`

Example:

Event::NewServer::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: id=
693EF360-B908-11DC-9EA5-10BF483EF417, ip=172.16.2.22, state=initialization

OtherServerStateChange:

Message: `ip=<ipAddr>, state=<from>-><to>`

Example:

Event::OtherServerStateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-
02:42:56:PM::15::: ipAddress=169.254.1.10->172.16.2.22, state=slave->master

DeviceStatus: initial device state

Message: `state=down`
Message: `state=up, uptime=<seconds>, cable=disconnected`
Message: `state=up, uptime=<seconds>, cable=connected, sizeX=<pixels>, sizeY=<pixels>, fps=<fps>`
Message (DECODER): `state=up, uptime=<seconds>, cable=connected, sizeX=<pixels>, sizeY=<pixels>, fps=<fps>, receivingVideoFromEncoder=no|yes|yes-with-warning [, reason=<reason> | warning=<warning>]`

"no" reason list:

decoder not joined
decoder down
decoder hdmi down
encoder down
encoder stream disabled
encoder hdmi down
display does not support resolution
encoder and decoder hdcp do not match
encoder has unsupported color format
encoder data rate exceeded
decoder data rate exceeded
multiview error -- do 'show multiviews status'
decoder resolution < UHD; can't join multiview
encoder resolution < UHD; can't join video-wall
encoder has multiview conflict with genlock
problem with network connection

"yes-with-warning" warning list:

multiview partially active -- do 'show multiviews status'

Examples:

Event::DeviceState::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=down
Event::DeviceState::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=up, uptime=1234, cable=connected, sizeX=3840, sizeY=2160, fps=60,
receivingVideoFromEncoder=no, reason=encoder hdmi down

DeviceStateChange: device up/down

Message: `state=up|down`
Example:

Event::DeviceStateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=up

CableConnection

Message: `cable=connected|disconnected`
Example:

Event::CableConnection::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
cable=connected

VideoStatusChange: sending video or not and why

Message: `receivingVideoFromEncoder=no|yes|yes-with-warning [, reason=<reason> | warning=<warning>]`

Example:

Event::VideoStatusChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
receivingVideoFromEncoder=no, reason=encoder hdmi down

ResolutionChange: just resolution change

Message: `sizeX=from->to, sizeY=from->to, fps=from->to`

Example:

Event::ResolutionChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
sizeX=1920->3840, sizeY=1080->2160, fps=59.95->59.94

VideoChanged: encoder only, non-resolution change

Message: `interlaced=<from>-><to>, color=<from>-><to>, colorDepth=<from>-><to>, hdcp=<from>-><to>, hdcpVersion=<from>-><to>, hdmi20=<from>-><to>`

Where:

Interlaced: yes, no
Color values: RGB, YUV444, YUV422, YUV420
colorDepth: 8, 10, 12
hdcp: yes, no
hdcpVersion: none, 1.4, 2.2
hdmi20: yes, no

Example:

Event::VideoChanged::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
interlaced=yes->no, color=RGB->YUV444, colorDepth=8->10, hdcp=yes->no, hdcpVersion=1.4->2.2, hdmi20=no->yes

RS232Data

Message: `data=<rs232Data>`

Example:

Event::RS232Data::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
data="hello there"

Note: RS232 events are only sent after a termination character has been received. If there are no termination characters defined, an event is generated after 10ms of no additional input.

IRData

Message: `data=<irData>`

Example:

Event::IrData::DCD(d8:80:39:9a:d0:cd)::Jun-19-05:02:07:PM::133::: data=
\"0000006d0000002700ae00a70016000f001500100015000f001500100015000f001
500340016000f0015000f0015000f0015001000140010001400100015000f0014001000-
8001600350016003400160034001600340016000f001500100015000f001500100015000f0015001000
1400100014001000150 00f0015003400160033001600340016003400160033001600330016 0-
034001600330016017c\"

AutoEdidSelect

Message: `sourceDecoder=<dec>, edidValid=yes|no, pixClockMhz=<from>-><to>, color=<from>-><to>, colorDepth=<from>-><to>, onlyPcmAudio=<from>-><to>`

Example:

Event::AutoEdidSelect::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
sourceDecoder=D14, edidValid=yes, pixClockMhz=150->600, color=RGB->YUV420, colorDepth=8->10, onlyPcmAudio=no->no

MulticastConflict

Message: `conflict=<multicastAddr>, action=getting new address`

Example:

Event::MulticastConflict::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
conflict=224.1.1.1, action=getting new address

AllocMcastFailed

Message: `allocation=failed`

Example:

Event::AllocMcastFailed::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
allocation=failed

EdidFirstChecksumInvalid

Message: `firstChecksum=invalid`

Example:

Event::EdidFirstChecksumInvalid::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: firstChecmsum=invalid

EdidSecondChecksumInvalid

Message: `secondChecksum=invalid`

Example:

Event::EdidSecondChecksumInvalid::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: secondChecmsum=invalid

IRDongleButtons

Message: `data=<irDongleButtons>`

Note this event is closely related to the IrData event. If the IrData "data" matches known pattern for IrDongleButton, it will trigger the IRDongleButtons event as shown below.

Examples:

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:26:PM::28::: button0=open, button1=open

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:32:PM::30::: button0=open, button1=closed

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:35:PM::31::: button0=open, button1=open

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:38:PM::32::: button0=open, button1=open

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:41:PM::33::: button0=open, button1=closed

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:44:PM::34::: button0=closed, button1=closed

Event::IrDongleButtons::d33(d8:80:39:eb:39:33)::Jan-12-06:27:47:PM::35::: button0=open, button1=closed

5 SNMP and LLDP

ZyPer Management Platform SNMP and LLDP support

The ZyPer Management Platform provides SNMPv2c and SNMPv3 access to status and configuration for the server and all devices. Specifically, it provides access to:

- Server information and configuration
- General device status and configuration (common to encoders and decoders)
- Decoder-specific status and configuration
- Encoder-specific status and configuration

The SNMP MIB, ZEEVEE-MIB.txt, may be obtained from www.zeevee.com. The MIB (Management Information Base) provides variable definitions for the above data. Note that ZEEVEE-MIB.txt imports several other standard MIBs, which are also available from the ZeeVee website for convenience.

The ZMP SNMP implementation also supports the following standard MIBs for generic host management:

- SNMPv2-MIB
- IF-MIB
- IP-MIB
- TCP-MIB
- UDP-MIB

And finally, a very common companion to SNMP is LLDP – Link Layer Discover Protocol. ZMP runs an LLDP daemon that collects link neighbor information. That information may be retrieved using SNMP to create a network topology map. ZMP supports the standard MIB, LLDP-MIB.txt, to enable this.

SNMP Agent SNMPv3 Account Configuration

ZMP SNMPv3 implements a simplified User-based Security Model (USM). Client accounts use MD5 authentication without encryption and are only read-only.

Further, ZMP effectively does not implement the View-based Access Control Model (VACM), in that each account has access to the entire mib.

From the ZMP CLI, the following commands manage SNMPv3 accounts:

```
add snmp user v3 accessLevel readOnly encrypted no
username <newSnmpUser> password <string>
delete snmp user v3 username <string>
show snmp users
```

For example, to add user testAccount, with a password, myPassword, the following would be entered:

```
Zyper$ add snmp user v3 accessLevel readOnly auth
MD5 encrypted no username testAccount password
myPassword

Zyper$ show snmp users
snmp(172.16.2.169);
  snmp.user: version=v3, auth=MD5, encryption=none,
  username=testAccount
```

SNMP Agent SNMPv2c Account Configuration

SNMPv2c accounts are based only on a username, although SNMP calls it a "community". There is no secure authentication of the user and no encryption.

As with V3, ZMP effectively does not implement VACM – that is, each user/community has access to the entire mib.

From the ZMP CLI, the following commands manage SNMPv2c accounts:

```
add snmp user v2c accessLevel readOnly community
<string>
delete snmp user v2c comunity <snmpUser>
show snmp users
```

For example:

```
Zyper$ add snmp user v2c accessLevel readOnly
community public

Zyper$ show snmp users
snmp(172.16.2.169);
  snmp.user: version=v3, auth=MD5, encryption=none,
  username=testAccount
  snmp.user: version=v2c, community=public
```

SNMP Client

Choose an SNMP client. There are many. The following examples use the client applications provided in the Linux Net-SNMP package.

Examples using Linux Net-SNMP and the above V3 account to retrieve the "zvzServerInfo" group of variables defined in the ZEEVEE-MIB.txt file:

```
        bin $ snmpwalk -u testAccount -l authNoPriv  -A
myPassword 172.16.2.169 zvzServerInfo
        ZEEVEE-MIB::zvzServerInfoHostname.0 = STRING: rey
        ZEEVEE-MIB::zvzServerInfoVersion.0 = STRING: 2.1.35413:3
5414
        ZEEVEE-MIB::zvzServerInfoPreviousVersion.0 = STRING: N/A
        ZEEVEE-MIB::zvzServerInfoSerialNumber.0 = STRING:
ZZMPFB0002b3913cf8A
        ZEEVEE-MIB::zvzServerInfoUptime.0 = Timeticks: (0) 0:00:
00.00
        ZEEVEE-MIB::zvzServerInfoLicenseLimit.0 = INTEGER: 0
        ZEEVEE-MIB::zvzServerInfoKnownDevices.0 = INTEGER: 12
        ZEEVEE-MIB::zvzServerInfoDevicesUp.0 = INTEGER: 7
        ZEEVEE-MIB::zvzServerInfoDevicesExceeded.0 = INTEGER: 0
```

And a similar example using the V2c account created above:

```
        bin $ snmpwalk -v2c -c public 172.16.2.169
zvzServerConfig
        ZEEVEE-MIB::zvzServerConfigAutoEdidMode.0 = INTEGER:
enabled(1)
        ZEEVEE-MIB::zvzServerConfigHdmiAudio.0 = INTEGER:
allowCompressed(1)
        ZEEVEE-MIB::zvzServerConfigLicense.0 = STRING: none
```

SNMP Notifications

There are four different types of SNMP Notifications. The original was for SNMPv1. That format is no longer in use. Then came v2C, which is very easy to use, but not used often due to lack of authentication or encryption. SNMPv3 has two different notification mechanisms. They both use the same format as defined in SNMPv2c, but add authentication and encryption. The first v3 notification is stilled called a trap. It is similar to v2c in that it is unreliable – the trap sender sends it over UDP and forgets. The configuration required for this form of notification is surprisingly terrible, and for that reason, not supported by ZMP. The final notification form is called an "Inform". This is the same as the trap, but reliably sent. The receiver must acknowledge the inform, and the send must try a number of times until an acknowledgment is received. Interestingly, using the Inform is not very hard.

NOTE: Roles are reversed for notifications! ZMP is the client and your trap server… is the server. That means that to receive either v2c traps, or v3c informs, you must configure your trap daemon with the correct authentication and access control. As a warning, the net-snmp trap daemon, as with its snmp daemon, is just bit tricky to use.

From the ZMP CLI, the following commands manage notification generation:

```
    add snmp trapServer v2cTrap ipAddress <address:ip> community
<string>
        add snmp user v2c accessLevel readOnly community
<newSnmpCommunity>
        add snmp user v3 accessLevel readOnly auth MD5 encrypted no
username <newSnmpUser> password <string>
        delete snmp trapServer v2cTrap <address:ip> community
<string>
        delete snmp user v2c community <snmpCommunity>
        delete snmp user v3 username <snmpUser>
        show snmp trapServers
        show snmp users
```

More Techinical Information on the Linux Net-SNMP Package

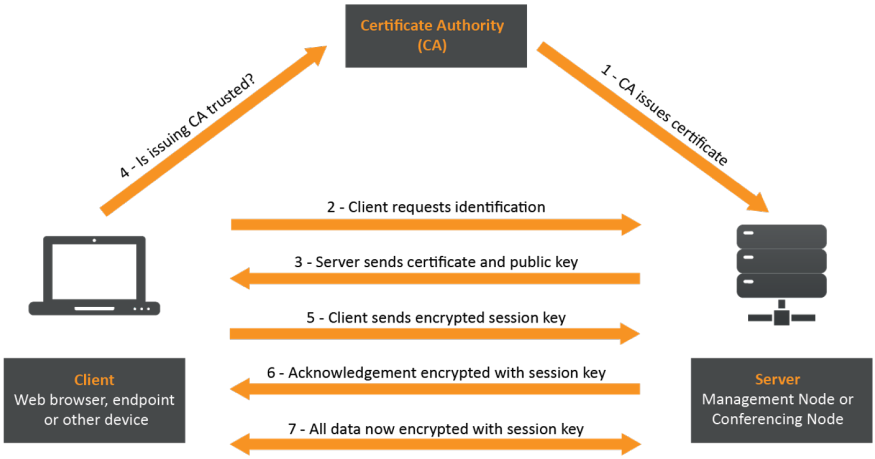
If NetSNMP is installed on Ubuntu Linux, then you either have to install the mibs in /usr/share/snmp/mibs, or use the -M command line switch to *fully* specify the path (cannot use the ~ character). For example, if the mibs are placed under you home directory, "snmp-mibs", the following will work:

```
snmpwalk -M +${HOME}/snmp-mibs  -m all -u testAccount -l
authNoPriv  -A myPassword 127.0.0.1 zvzServerInfo
```

Note the above example uses the account and password created in the earlier example. Further, if the mibs are copied to the /usr/share/snmp/mibs directory, then the -M in the above command may be omitted.

Note also that the Ubuntu net-snmp directory structure does not follow the man pages for net-snmp apps. Very annoying. However, any ZMP system (including in-house servers and build systems), the "standard" directories are followed. In this case, the mib directory is /usr/local/share/snmp/mibs.

6 TLS 1.3 Support



ZyPer Management Platform TLS 1.3 support

There are two parts to TLS 1.3 support:

- web server tls certificate request and install
- certificate authority

In most cases, only the web cert "request and install" functionality is used along with some external CA (like Verisign). However some sites will find the internal CA very convenient. And it makes testing far easier too.

New API input feature:

-- Prompt for input for no-echo and multi-line input

-- If a command line field allows an asterisk (*), the user will be prompted for input after a CR

-- Multiline input is terminated with a control-d

-- In some cases input may be on the command line OR prompted for after the CR

Command list:
-- Don't forget that "help subsystem tls" will show you the following!

generate tls ca privKeyPass <string>|* country <string> state <string> locality <string> organization <string> organizationUnit <string> email <string>
generate tls server csr privKeyPass <string>|* fqdn <string> country <string> state <string> locality <string> organization <string> organizationUnit <string> email <string>
generate tls device csr privKeyPass <string>|* fqdn <string> country <string> state <string> locality <string> organization <string> organizationUnit <string> email <string>
load tls ca cert fromInput *
load tls ca cert fromFile <certsFilename>
load tls ca privateKey privKeyPass <string>|* fromInput *
load tls ca privateKey privKeyPass <string>|* fromFile <certsFilename>
load tls server caIntermediates fromInput none|*
load tls server caIntermediates fromFile <certsFilename>|none
load tls server cert fromInput *
load tls server cert fromFile <certsFilename>
load tls server privateKey privKeyPass <string>|* fromInput *
load tls server privateKey privKeyPass <string>|* fromFile <certsFilename>
load tls device caIntermediates fromInput none|*
load tls device caIntermediates fromFile <certsFilename>|none
load tls device cert fromInput *

```
load tls device cert fromFile <certsFilename>
load tls device privateKey privKeyPass <string>|* fromInput *
load tls device privateKey privKeyPass <string>|* fromFile <certsFilename>
set tls server mode enabled|disabled
set tls server fqdn <string>|fromCert
show tls summary
show tls pem server csr
show tls pem server cert
show tls pem server privKey
show tls pem server caIntermediates
show tls device summary
show tls pem device csr
show tls pem device cert
show tls pem device privKey
show tls pem device caIntermediates
show tls pem ca cert
show tls pem ca privKey
show tls pem ca signedCert
sign tls csr caPrivateKeyPass <string>|* fromInput *
sign tls csr caPrivateKeyPass <string>|* fromFile <certsFilename>
```

To start

```
-- Show overall tls info
   Zyper$ show tls summary
   server(192.168.0.22);
    server.tls.server; tlsMode=disabled, fqdn=NA
    server.tls.csr; status=invalid
    server.tls.serverCert; status=invalid
    server.tls.caChainCert; status=invalid
    server.tls.caCert; status=invalid
    server.tls.signed; status=invalid
   Success
   Zyper$
```

Tests

1. Set up the internal CA

```
-- Create CA private key and (root) certificate
   Zyper$ generate tls ca privKeyPass * country US state MA locality littleton
organization jagCerts organizationUnit money email me@there.com
      -- privKeyPass: if a *, then prompted after CR.
      -- a passphrase is required
```

- - - -

```
-- show resulting private key
    Zyper$ show tls server pem privKey
    pemData:
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: AES-256-CBC,0503C6FA20C8D8B313AECD56646B8634
    ...
    -----END RSA PRIVATE KEY-----

-- Show summary
    Zyper$ show tls summary
    server(192.168.0.22);
     server.tls.server; tlsMode=disabled, fqdn=NA
     server.tls.csr; status=valid, C=us, CN=jag-zmp-pro.zeevee.com, O=z,
OU=d, ST=ma, emailAddress=j@z
     server.tls.serverCert; status=invalid
     server.tls.caChainCert; status=invalid
     server.tls.caCert; status=valid, C=US, CN=caCert, L=littleton, O=jagCerts,
OU=money, ST=MA, emailAddress=me@there.com
     server.tls.caCert; issuer=caCert
     server.tls.caCert; fingerprint=D44C2B45F75435DC95A618F800C7AB1E458
4264B
     server.tls.caCert; expires=06/07/32T17:31:53-0400
     server.tls.signed; status=invalid
    Success
```

3. Use CSR to create signed certificate

```
  -- Copy the CSR data from the above "show tls server pem csr" command
  -- Include the ----BEGIN and ----END lines
  -- Issue the sign command:
    Zyper$ sign tls csr caPrivateKeyPass * fromInput *
    Enter passphrase: *******                <<<< USER INPUT CA private key
passphrase
    Enter PEM text (ctr-d to end):
    -----BEGIN CERTIFICATE REQUEST-----          <<<< USER INPUT(pastes)
CSR pem data
    ...
    -----END CERTIFICATE REQUEST-----
                                  <<<< USER INPUT control-d to end
    Success
    Zyper$

  -- Show signed certificate
    Zyper$ show tls ca pem signedCert
    pemData:
    -----BEGIN CERTIFICATE-----
    ...
    -----END CERTIFICATE-----
```

```
-- Show summary
    Zyper$ show tls summary
    server(192.168.0.22);
      server.tls.server; tlsMode=disabled, fqdn=NA
      server.tls.csr; status=valid, C=us, CN=jag-zmp-pro.zeevee.com, O=z,
OU=d, ST=ma, emailAddress=j@z
      server.tls.serverCert; status=invalid
      server.tls.caChainCert; status=invalid
      server.tls.caCert; status=valid, C=US, CN=caCert, L=littleton, O=jagCerts,
OU=money, ST=MA, emailAddress=me@there.com
      server.tls.caCert; issuer=caCert
      server.tls.caCert; fingerprint=D44C2B45F75435DC95A618F800C7AB1E458
4264B
      server.tls.caCert; expires=06/07/32T17:31:53-0400
      server.tls.signed; status=valid, C=us, CN=jag-zmp-pro.zeevee.com, O=z,
OU=d, ST=ma, emailAddress=j@z
      server.tls.signed; issuer=caCert
      server.tls.signed; fingerprint=B6A6F05F1CA0486E15BCA969472EF6247E2F
6C80
      server.tls.signed; expires=06/10/23T17:42:37-0400
```

4. Load signed certificate

```
  -- In this case, from our above internal CA generated cert.
  -- But in other cases, some external CA will provide the signed cert (same
PEM file format)

  -- Copy cert from previous "show tls ca pem signedCert"
     -- Or from external CA generated pem data
  -- Issue command:
     Zyper$ load tls server cert fromInput *
     Enter PEM text (ctr-d to end):
     -----BEGIN CERTIFICATE-----              <<< USER INPUT(paste) pem data
for signed cert
     -----END CERTIFICATE-----
                                   <<< USER INPUT control-d to end
```

```
-- Show summary:
    Zyper$ show tls summary
    server(192.168.0.22);
    server.tls.server; tlsMode=disabled, fqdnMode=fromCert, fqdn=jag-zmp-
pro.zeevee.com
    server.tls.csr; status=valid, C=US, CN=jag-zmp-pro.zeevee.com, O=ZeeVee,
OU=Money, ST=MA, emailAddress=me@there.com
    server.tls.serverCert; status=valid, C=US, CN=jag-zmp-pro.zeevee.com,
O=ZeeVee, OU=Money, ST=MA, emailAddress=me@there.com
    server.tls.serverCert; issuer=caCert
    server.tls.serverCert; fingerprint=1A27CD38AF2EC44693934BE99D016A2D8
5194F47
    server.tls.serverCert; expires=03/20/24T09:39:49-0500
    server.tls.caChainCert; status=invalid
    server.tls.caCert; status=valid, C=US, CN=caCert, L=Litteton, O=ZeeVee,
OU=Money, ST=MA, emailAddress=me@there.com
    server.tls.caCert; issuer=caCert
    server.tls.caCert; fingerprint=4EDC0BEFC191B5EBC26837ADAE775966C7A5
033E
    server.tls.caCert; expires=03/18/33T09:36:25-0500
    server.tls.signed; status=valid, C=US, CN=jag-zmp-pro.zeevee.com,
O=ZeeVee, OU=Money, ST=MA, emailAddress=me@there.com
    server.tls.signed; issuer=caCert
    server.tls.signed; fingerprint=1A27CD38AF2EC44693934BE99D016A2D8519
4F47
    server.tls.signed; expires=03/20/24T09:39:49-0500
 Success

 -- You can see that the serverCert is valid
```

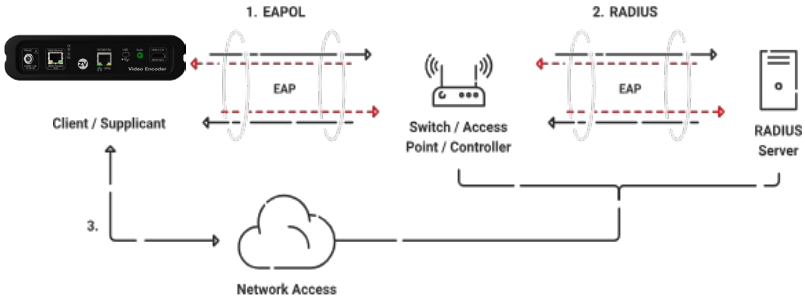
5. Enable web server tls mode

```
    Zyper$ set tls server mode enabled
    Success
```

NOTE: The cert defines the cert contents (asn.1) and there are several encodings: PEM, DER and PKCS#7/CMS

- - - -

7 802.1X Support



ZyPer4K-XS/XSE and ZyPer4K-XR 802.1X Support

Important Note: ZyPer4K-XS/XSE or XR units must have firmware version 2.1.0.1 or newer installed to support the 802.1X feature.

Procedure to enable 802.1X

Enable server-device secure communication

1. set server security deviceSecurity <key>
2. set device <XS/XR Device> security enabled

Create CA Certificate

Similar to the process for enabling TLS support, you need to create a Certificate Authority (CA Cert) if you have not already done so. The same CA Cert can be used for the Server Cert as there is only one CA in the ZMP.

Example Command

generate tls ca privKeyPass * country US state MA locality Billerica organization ZeeVee organizationUnit Money email aweeks@zeevee.com

Enter passphrase: ******

Success

Generate device CSR, sign it and load it

Example Commands

1. generate tls device csr privKeyPass * fqdn zeevee.com country US state MA locality Billerica organization ZeeVee organizationUnit Money email aweeks@zeevee.com

Enter passphrase: ***

Success

2. show tls pem device csr

show tls pem device csr
\

pemData:

-----BEGIN CERTIFICATE REQUEST-----

MIIBbjCCARQCAQAwcjEgMB4GCSqGSIb3DQEJARYRYXdlZ.....

.....xhimQ/NNCikyXvOrmLGTGg==

-----END CERTIFICATE REQUEST-----

Success

2a. show tls device summary -- to see valid csr

show tls device summary

server(192.168.0.21);

device.tls.csr; status=valid, C=US, CN=zeevee.com, O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com

device.tls.deviceCert; status=invalid

device.tls.deviceCaChainCert; status=valid, C=US, CN=caCert, L=Billerica, O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com

device.tls.deviceCaChainCert; issuer=caCert

device.tls.deviceCaChainCert; fingerprint=A7C444401F258372BEB6E015EE1FC4E357AB4E89

device.tls.deviceCaChainCert; expires=01/06/34T15:24:05-0500

device.tls.deviceCaCert; status=invalid

Success

3. sign tls csr caPrivateKeyPass PASSWORD fromInput *

Copy from step 2, paste, control-d

sign tls csr caPrivateKeyPass * fromInput *

Enter passphrase: ******

Enter PEM text (ctr-d to end):

-----BEGIN CERTIFICATE REQUEST-----

MIIBbzCCARQCAQAwcjEgMB4GCSqGSIb3DQEJARYR...

AiEAxOR23t5VL8qvzAodFBfqYfRknP4wj4+v2PcemglOe8s=

-----END CERTIFICATE REQUEST-----

Success

4. show tls pem ca signedCert

show tls pem ca signedCert

pemData:

-----BEGIN CERTIFICATE-----

MIIEFzCCAf+gAwIBAgICEjQwDQYJKoZIhvcNAQELBQAwgYIxIDAeBgkqhkiG9w...
5Wn9zzv5oMQoyoOMeZU1K+V2ONJtUBnS2vwINTmw8HdiBY6RsuK4w==

-----END CERTIFICATE-----

Success

5. load tls device cert fromInput *

Copy from step 4, paste, control-d

load tls device cert fromInput *

Enter PEM text (ctr-d to end):

-----BEGIN CERTIFICATE-----

MIIEFzCCAf+gAwIBAgICEjQwDQYJKoZIhvcNAQELBQAwgYIxIDAeBgkqhkiG9w...

5Wn9zzv5oMQoyoOMeZU1K+V2ONJtUBnS2vwINTmw8HdiBY6RsuK4w==

-----END CERTIFICATE-----

Success

5a. show tls device summary -- to see valid csr AND valid cert

OPTIONAL

show tls device summary

server(192.168.0.21);

device.tls.csr; status=valid, C=US, CN=zeevee.com, O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com

device.tls.deviceCert; status=valid, C=US, CN=zeevee.com, O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com

device.tls.deviceCert; issuer=caCert

device.tls.deviceCert; fingerprint=44CD6AB55BAB13AF3B7A76E453A307761FDBCB6D

device.tls.deviceCert; expires=01/11/25T13:58:19-0500

device.tls.deviceCaChainCert; status=valid, C=US, CN=caCert, L=Billerica, O=ZeeVee, OU=Money, ST=MA, emailAddress=aweeks@zeevee.com

device.tls.deviceCaChainCert; issuer=caCert

device.tls.deviceCaChainCert; fingerprint=A7C444401F258372BEB6E015EE1FC4E357AB4E89

device.tls.deviceCaChainCert; expires=01/06/34T15:24:05-0500

device.tls.deviceCaCert; status=invalid

Success

6. load tls device caIntermediates fromInput *

load tls device caIntermediates fromInput *

Enter PEM text (ctr-d to end):

-----BEGIN CERTIFICATE-----

MIIEFzCCAf+gAwIBAgICEjQwDQYJKoZIhvcNAQELBQAwgYIxIDAeBgkqhkiG9w...

5Wn9zzv5oMQoyoOMeZU1K+V2ONJtUBnS2vwINTmw8HdiBY6RsuK4w==

-----END CERTIFICATE-----

Success

802.1X should now be available on the interface.

Here's a summary of usage sequences…

If using ZMP CA:

Generate CA certificate

If running ZMP securely with TLS:

Generate ZMP csr

If using ZMP CA, copy CSR to CA panel to be signed

Otherwise use external CA to sign

Load signed cert into ZMP TLS panel

Enable server TLS mode

If running 802.1x EAP-TLS on devices

Enable zmp-device security on all devices (set server key and enable on all devices)

Generate radius CSR

Download private key to be used by radius server

If using ZMP CA, copy CSR to CA panel to be signed

Otherwise use external CA to sign

Load signed cert and private key into radius server

Generate device csr

If using ZMP CA, copy CSR to CA panel to be signed

Otherwise use external CA to sign

Load signed cert into 802.1x TLS panel, device cert

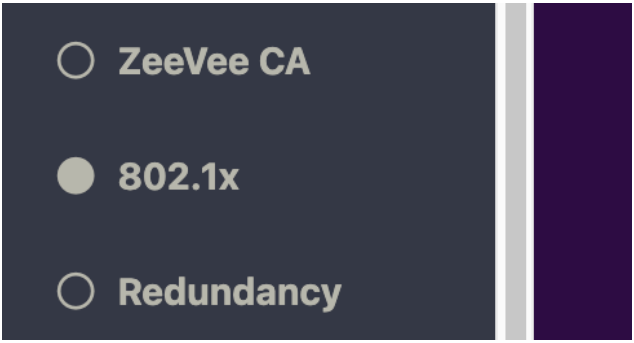
Please note that both the Network Switch and RADIUS Server will need to be setup and configured as well. Please consult switch manufacturer for appropriate documentation.

802.1X Information in the ZMP Browser GUI

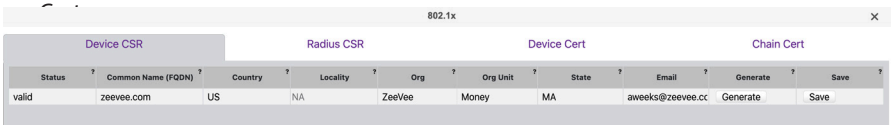
802.01 Information can also be found in the ZMP Browser GUI.

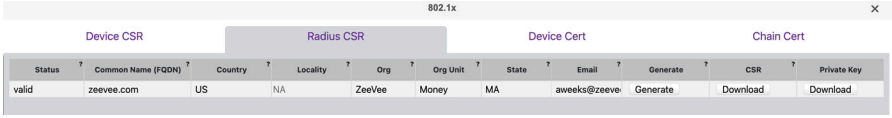
1. Login to the ZMP. Refer to Accessing ZyPer Management Platform (page 13) for more information.

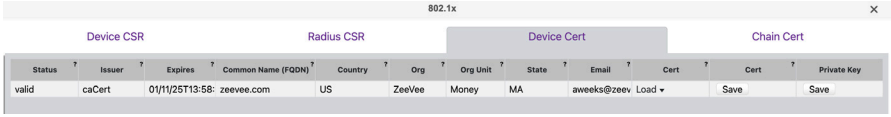
2. Click the 802.1X tab at the left of the page.

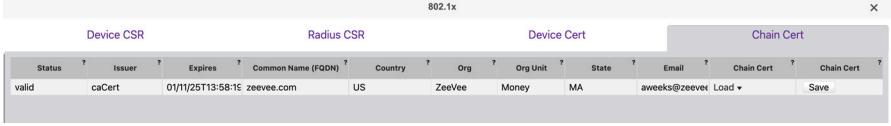


3. Just as was detailed above using the Command Line Interface (CLI) you can use the browser GUI to Generate the Device CSR, Radius CSR, Device Cert and Chain Cert.









8 Active Directory Lightweight Directory Services

AD LDS

Lightweight Directory Services

Configuring Active Directory Lightweight Directory Services attributes for ZMP (Using ADSI Edit)

This section outlines the necessary attribute configurations on an ADLDS (Active Directory Lightweight Directory Services) server to enable seamless integration with a ZMP. ZMP uses SSSD (System Security Services Daemon) client which provides authentication and access control by leveraging LDAP and other identity management protocols.

Prerequisites

- An ADLDS server properly set up and accessible from the Linux client.
- Administrative access to modify LDAP attributes using **ADSI Edit**.

1. (AD) Create certificate to load on ZMP
2. (AD) Create an LDAP bind account (service account)

- Read Permissions on Directory Objects:
  - The bind account needs permission to read the directory information it's intended to access. By default, most AD LDS environments allow "Authenticated Users" to read general directory data, so often no additional permissions are required.
  - If you have customized permissions or restricted access, you may need to grant explicit Read permissions on specific organizational units (OUs) or containers.

- Log On Locally (if applicable):
  - If the bind account needs to log on interactively to the server itself (rather than just performing LDAP operations), it requires the "Log on locally" right.

- Service Account Best Practices (if applicable):
  - Set the account to "Password never expires" to avoid disruptions.
  - Restrict login to specific systems if the account is only intended for LDAP binding.
  - Apply "Deny log on locally" and "Deny log on through Remote Desktop Services" to harden the account.

3. (AD) Identify(create) groups that will have different access levels on the ZMP

4. (AD) Identify(create) users that will have access to the ZMP

5. (AD) Assign users to groups

6. (AD) Using ADSI Edit modify each User attributes

ZMP requires specific User attributes to function correctly. Due to ADLDS's lack of native POSIX attributes, some attributes need to be mapped manually. Below are the required mappings between ADLDS and ZMP:

a. uidNumber: start at 10000 to avoid conflicts

b. gidNumber: start at 10000 to avoid conflicts

c. unixHomeDirectory: /home/username

Modifying Attributes Using ADSI Edit

1. Open **ADSI Edit** on the ADLDS server.

2. Connect to the desired ADLDS instance by right-clicking **ADSI Edit** and selecting **Connect to**.

3. Navigate to the appropriate **OU** or **CN** where your users or groups are stored.

4. Right-click on the user object and select **Properties**.

5. Locate the attributes to modify (e.g., `uidNumber`, `gidNumber`).

6. Click **Edit** to change the attribute value or **Add** if the attribute is not present.

7. Enter the desired value and click **OK**.

8. Repeat the process for each required attribute.

7. (ZMP) Load ADLDS certificate

8. (ZMP) create new Roles to match the groups on the ADLDS server

9. (ZMP) map Roles to gidNumber

10. (ZMP) Configure LDAP settings

a. Server Address: FQDN of ADLDS server

b. Base Search DN: in the format OU=Users,DC=domain,DC=com

c. Username: Bind account in DN format cn=<user>,ou=users,dc=domain,dc=com

d. Password: Bind account password.

Configuring ADLDS Attributes for SSSD Integration on Linux (Using ADSI Edit)

This section outlines the necessary attribute configurations on an ADLDS (Active Directory Lightweight Directory Services) server to enable seamless integration with an SSSD (System Security Services Daemon) client on a Linux system. SSSD provides authentication and access control by leveraging LDAP and other identity management protocols.

1. Prerequisites

- An ADLDS server properly set up and accessible from the Linux client.
- SSSD installed and configured on the Linux machine
- Administrative access to modify LDAP attributes using **ADSI Edit**.

2. Key Attribute Mappings

SSSD requires specific attributes to function correctly. Due to ADLDS's lack of native POSIX attributes, some attributes need to be mapped manually. Below are the typical mappings between ADLDS and SSSD:

SSSD Attribute	ADLDS Attribute	Description
`ldap_user_uid_number`	`uidNumber`	User ID number, usually manually assigned. Start at 10000 to avoid conficts
`ldap_user_gid_number`	`gidNumber`	Group ID number, usually manually assigned. Start at 10000 to avoid conficts
`ldap_user_home_directory`	`unixHomeDirectory`	Home directory path on the Linux client.
`ldap_user_shell`	`loginShell`	User's login shell (e.g., `/bin/bash`).

3. Modifying Attributes Using ADSI Edit

1. Open **ADSI Edit** on the ADLDS server.

2. Connect to the desired ADLDS instance by right-clicking **ADSI Edit** and selecting **Connect to**.

3. Navigate to the appropriate **OU** or **CN** where your users or groups are stored.

4. Right-click on the user object and select **Properties**.

5. Locate the attributes to modify (e.g., `sAMAccountName`, `uidNumber`, `gidNumber`).

6. Click **Edit** to change the attribute value or **Add** if the attribute is not present.

7. Enter the desired value and click **OK**.

8. Repeat the process for each required attribute.

Configuring the ADLDS server with the appropriate attributes using **ADSI Edit** ensures compatibility with SSSD, enabling Linux clients to authenticate and retrieve user data effectively. Proper mapping and TLS configuration are crucial for secure and reliable integration.

VAMX2

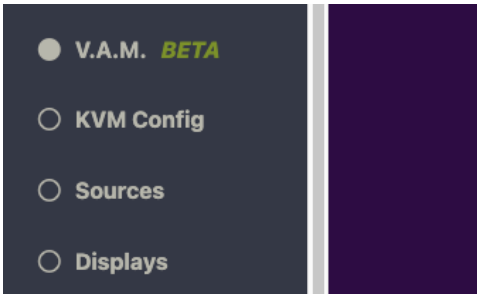
Visualize, Analyze, Monitor (V.A.M.)

This powerful tool can be used to Visualize, Analyze and Monitor multicast network traffic on the AV network and identify any potential issues where multicast traffic bandwidth may be greater than available bandwidth.

(Note this feature is available only on select manufactures switches.)

Currently Netgear and Cisco switches are supported in this "Beta" release of V.A.M.

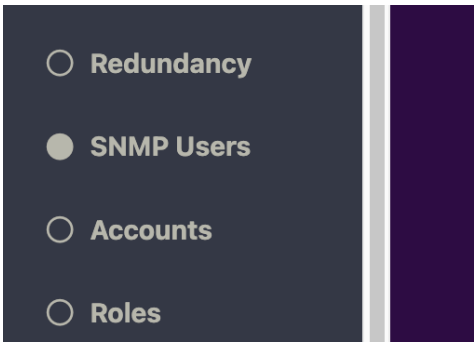
The V.A.M. Tool can be selected from on the left of the ZMP page.



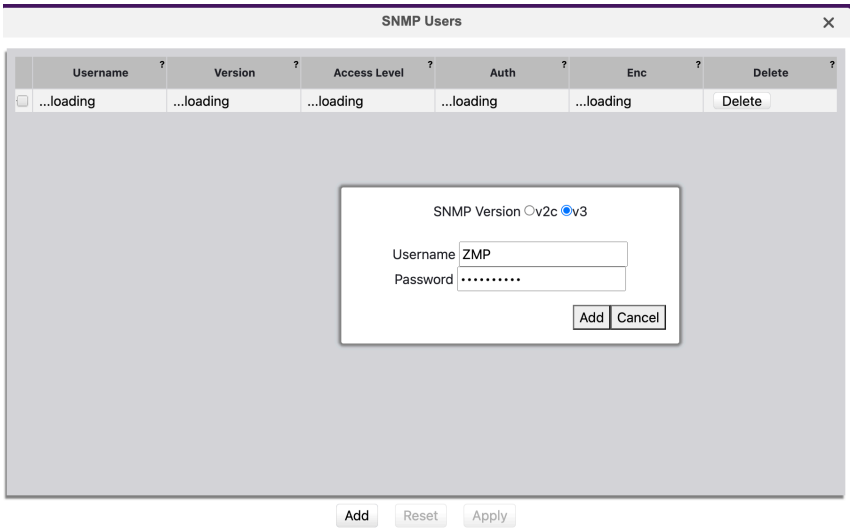
Configuration:

The ZyPer Mangement Platform itself will automatically be discovered by the V.A.M. tool. The first step is to Authorize the ZMP itself as a Netnode to provide detailed information back to the V.A.M. tool.

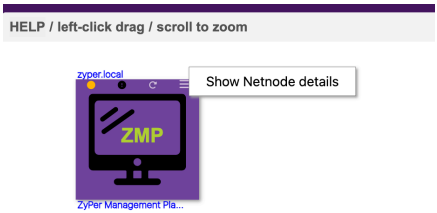
Open the SNMP Users tab to create an SNMP account for the ZMP itself.



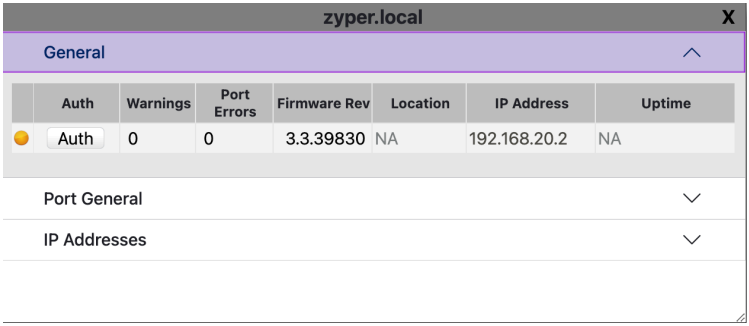
Click on the Add button and then create a Username and Password for the Netnode to be used with the ZMP. This can be just about anything you want. In this example I named it "ZMP"



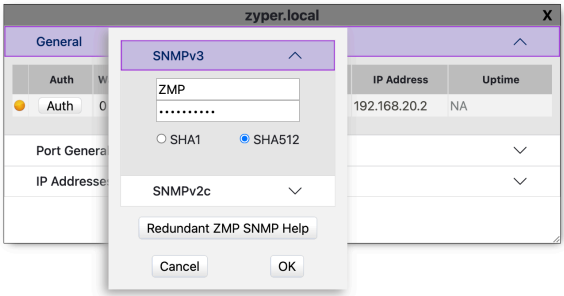
Now in V.A.M. window you can click on the 3 small bars in the ZMP to bring up a window to authorize the ZMP as a Netnode.



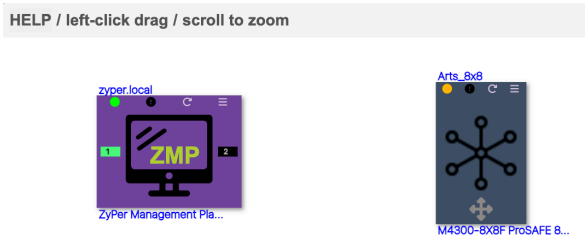
Click the Auth button to Authorize the Netnode



Enter the UserName and Password for the SNMP Netnode you created earlier.



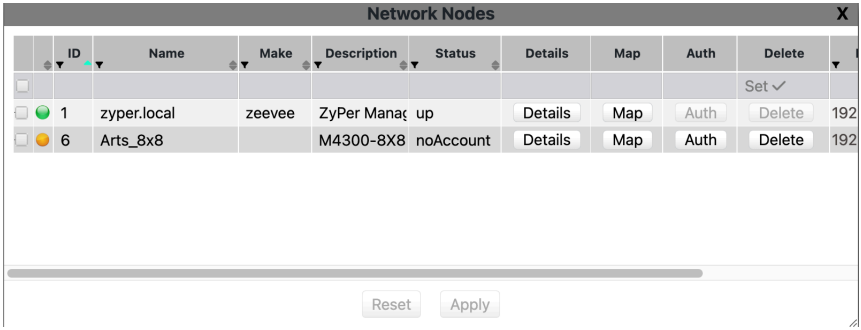
The dot in the upper left corner of the ZyPer Mangement Platform should now turn green.



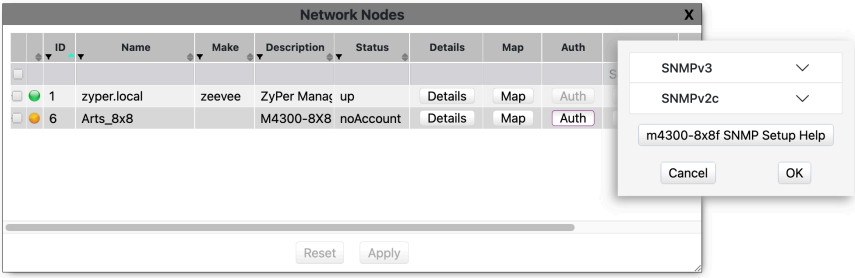
The network switch that is directly connected to the ZMP will automatically be discovered by the V.A.M. tool. The next step is to Authorize that switch to provide detailed information back to the V.A.M. tool.

Remember, the ZMP and connected switches are known as Network Nodes.

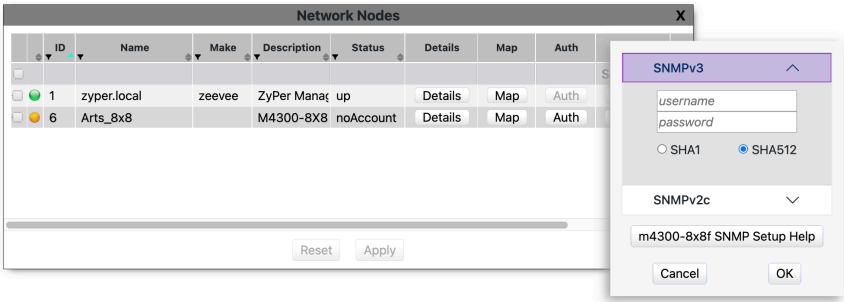
Click the 3 bars on the right side of the V.A.M. window to bring up the Network Nodes window.



Click on the "Auth" button for the Network Switch to initiate the Authentication process. In the example above, the switch is "Arts_8x8" and it is a Netgear M4300-8x8 switch.



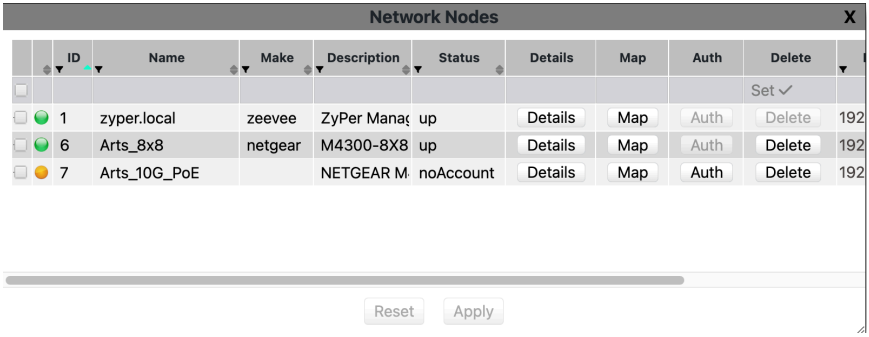
Choose the appropiate SNMP version for the switch. In this case with Netgear the proper choice is SNMPv3



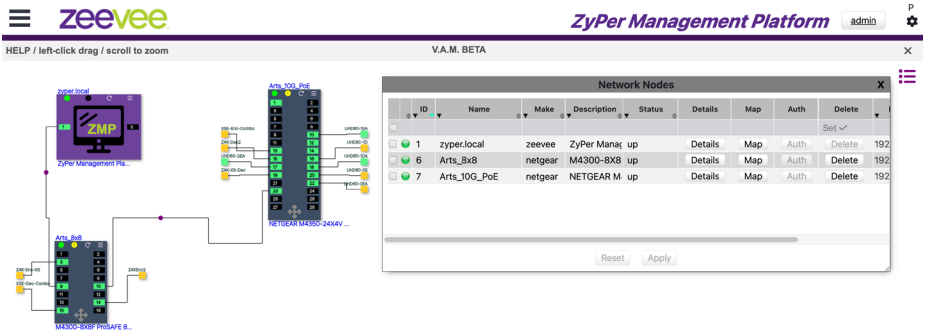
Enter the proper Username and Password for the network switch along with the correct selection for encryption. (SHA1 or SHA512)

For Netgear the proper selection is SHA512.

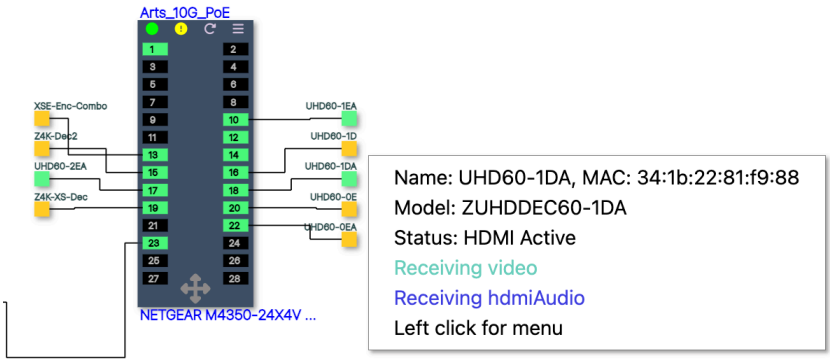
Once the first switch is authorized in the system, any additional switches connected to the first switch will now me discovered by V.A.M. Authorize additioinal switches one at a time.



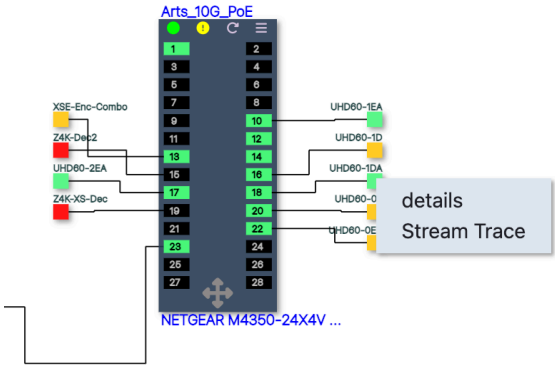
As you build out the Network Nodes, you will see a growing Visual representation of the network shown within the V.A.M. tool.



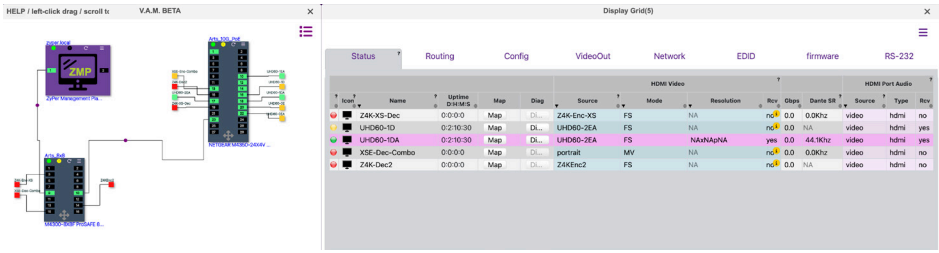
The Visual representation will show you the names and quick status of all ZyPer endpoints connected to the network. Hovering over a specific endpoint will allow you to bring up additional information. Below is example hovering over the endpoint named UHD60-1DA



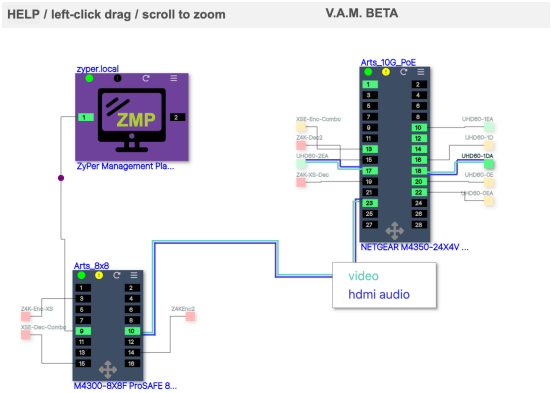
If you left click on an endpoint it will bring up small menu where you can select either "details" for that endpoint or show a graphical "stream trace" of multicase video and audio streams coming to or from that endpoint.



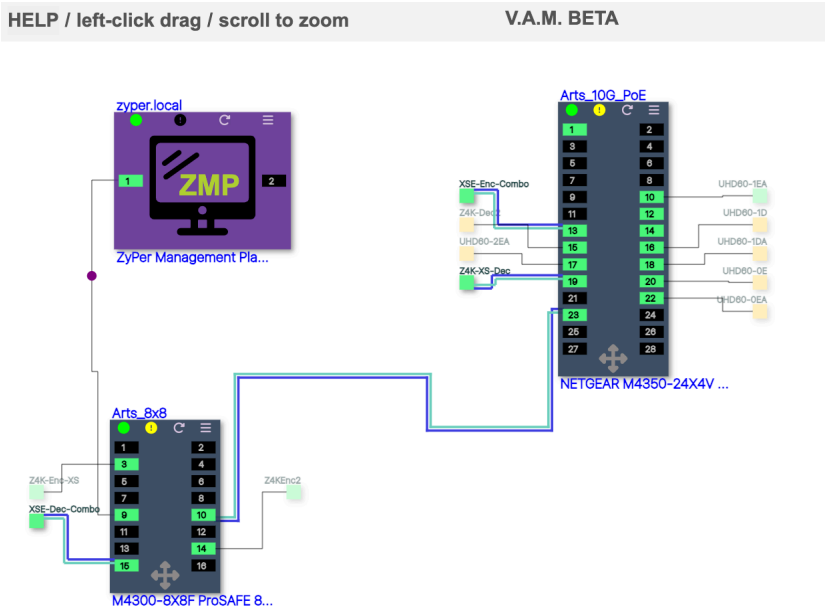
Clicking on "details" will open the Display or Source Grid Tab in the ZMP and highlight the specified endpoint. Below is example of clicking details for the UHD60-1DA endpoint. Notice UHD60-1DA is highlighted.



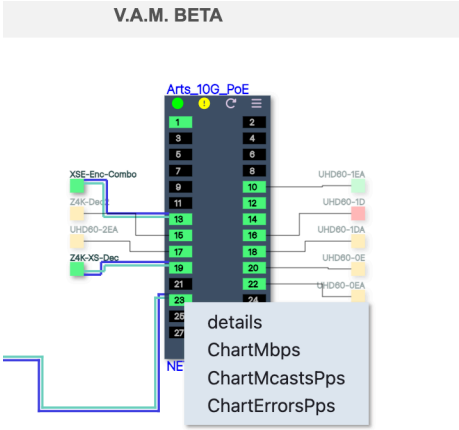
Clicking on Stream Trace will show a visual representation of where the video/ audio streams have come from or where they are going.



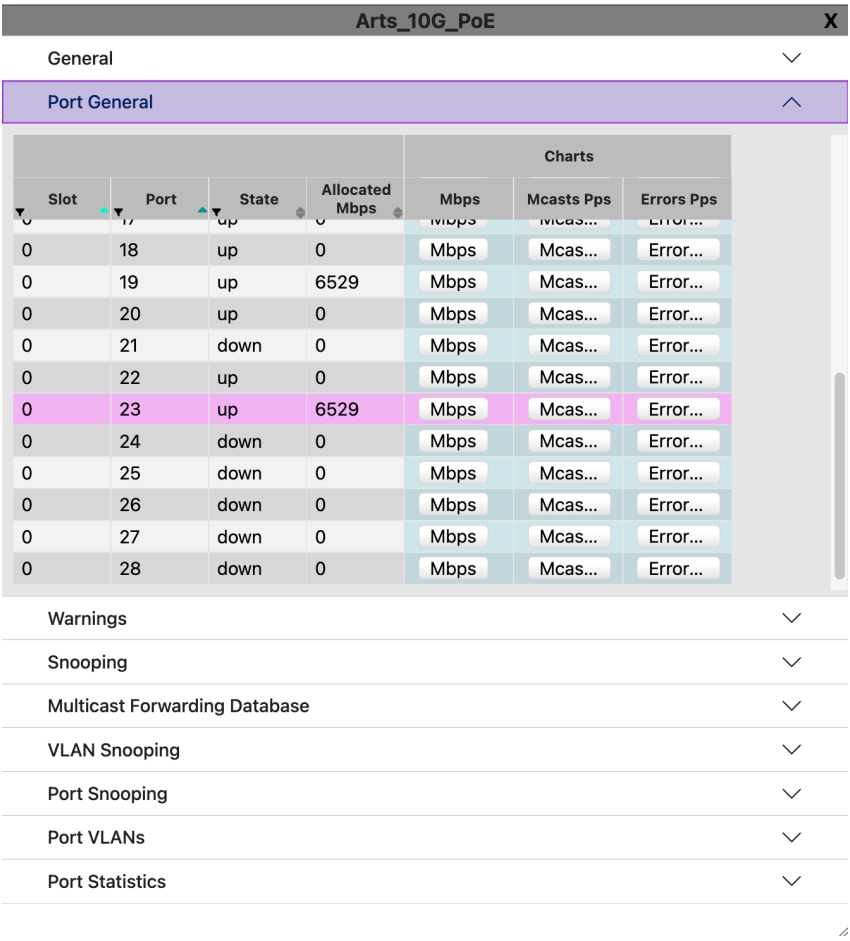
This example Stream Trace is from the Encoder XSE-Enc-Combo



If you left click on a Network port you are provided the following options:
Details, ChartMbps, ChartMcastsPps, ChartErrorsPps

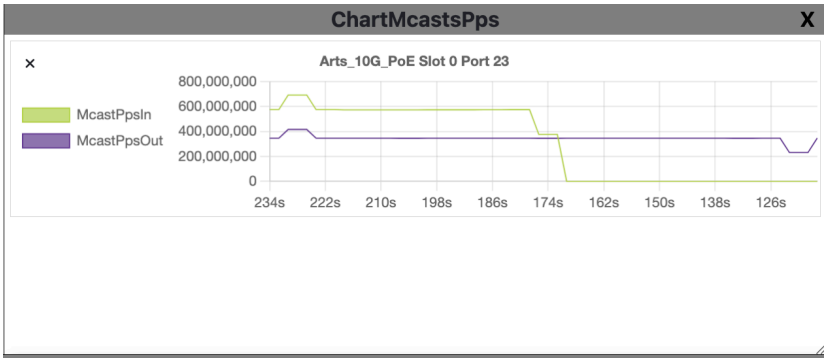


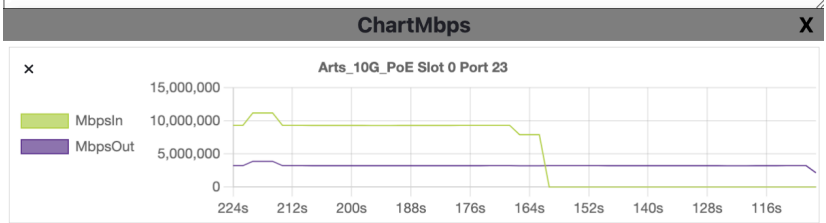
Clicking on Details will bring up detailed information provided by the network switch.

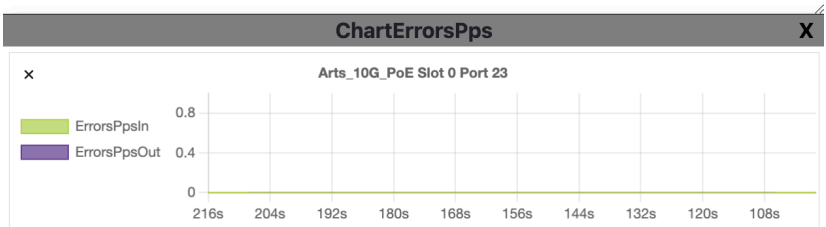


From here you can browse the various categories. General, Port General, Warnings, Snooping, Multicast Forwarding Database, VLAN Snooping, Port Snooping, Port VLANS and Port Statistics.

Clicking on any of the Charts will bring up the relevant chart. Below shows all 3 available charts for Port 23.

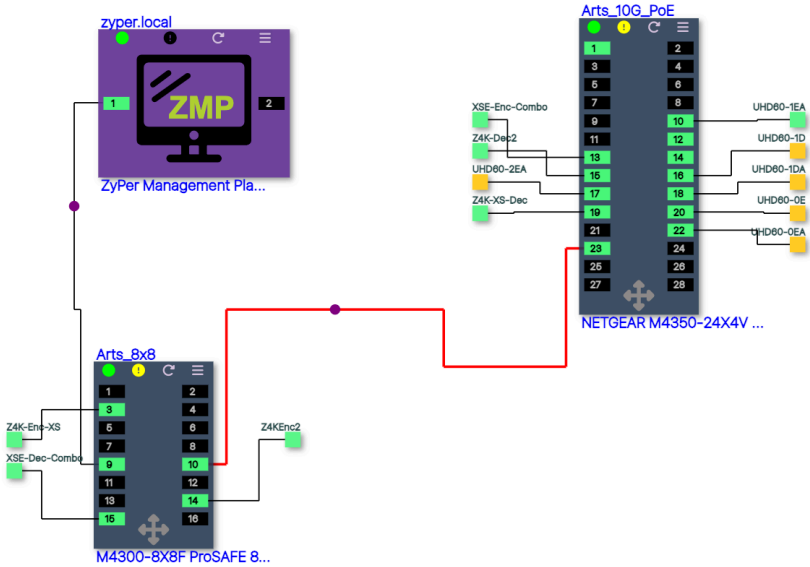






One of the features of V.A.M. is to visually notify the user when a port link becomes overloaded. In this case the link line between the switches becomes highlighed in Red.

HELP / left-click drag / scroll to zoom



In the example above the quantiy of multicast traffic over the link has exceeded the availabe bandwidth of that link. The user is visually notified of the issue so that action can be taken to resolve the issue.

10 Appendix

Updating the Software

Using Mac OS X

1. Make sure the Management Platform is powered and is working correctly.

2. Download the latest software from the ZeeVee website. Make note of the location of where the software was downloaded.

3. Launch the Terminal app, found under the Applications > Utilities folder. By default, the current directory will be the Home directory.

```
Last login: Tue Mar 22 14:24:08 on console
Andrews-MacBook-Pro:~ Andrew$
```

4. Change the directory to the location of the downloaded software file. For example, if the software was downloaded to the Desktop, then change to the Desktop folder, as shown:

```
Last login: Tue Mar 22 14:24:08 on console
Andrews-MacBook-Pro:~ Andrew$ cd desktop
Andrews-MacBook-Pro:desktop Andrew$
```

5. Use the FTP protocol to login to the Management Platform. At the terminal prompt, type the following and press the [ENTER] key.

```
Andrews-MacBook-Pro:desktop Andrew$ ftp 192.168.1.6
```

6. Enter the user name and password. Use anonymous for the user name and use guest for the password. The password will not be echoed to the screen.

```
Andrews-MacBook-Pro:desktop Andrew$ ftp 192.168.1.6
Connected to 192.168.1.6
220 (vsFTPd 3.0.2)
Name (192.168.1.6:Andrew): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

7. Type `cd files` at the ftp prompt to change to the /files directory.

```
ftp> cd files
250 Directory successfully changed.
ftp>
```

8. Enter and run the `put` command, followed by the full name of the software file, as shown. Make sure to replace [version] with the version of the filename you are using. For example:

```
ftp> put update_nuc_3.0.38847.zyper
```

9. Press the [ENTER] key. Information similar to the following will be displayed.

```
local: update_nuc_3.0.38847.zyper remote: update_nuc_3.0.38847.
zyper
229 Entering Extended Passive Mode (|||35257|).
150 Ok to send data.
100% |***********************************|  6830 KiB   94.30
MiB/s    00:00 ETA
226 Transfer complete.
6994519 bytes sent in 00:00 (92.30 MiB/s)
```

10. Type the `exit` command to exit FTP.

```
ftp> exit
Andrews-MacBook-Pro:desktop Andrew$
```

11. Telnet to the Management Platform, as shown.

```
$ telnet 192.168.1.6
Trying 192.168.1.6...
Connected to 192.168.1.6
Escape character is '^]'.
zyper$
```

12. Use the `update` command to update the Management Platform. Once entered, the Management Platform will reboot and the software will be updated. Note that the connection will be lost, temporarily, during the update process.

```
zyper$ update server update_nuc_3.0.38847.zyper
Success

Server rebooting; connection will end
```

Using Windows

1. Make sure the Management Platform is powered and is working correctly.

2. Download the latest software from the ZeeVee website. Make note of the location of where the software was downloaded.

3. Open Chrome and enter the IP address of the Management Platform using the FTP protocol. For example:

   ```
   ftp://169.254.185.207
   ```

4. The /files folder will be displayed.

5. Drag-and-drop the latest software file to the /files folder.

6. Use the Telnet protocol to access the Management Platform API.

7. Use the `update` command to update the Management Platform. Once entered, the Management Platform will reboot and the software will be updated. Note that the connection will be lost, temporarily, during the update process.

   ```
   zyper$ update server update_nuc_3.0.38847.zyper
   Success

   Server rebooting; connection will end
   ```

Using ZyPer Management Platform

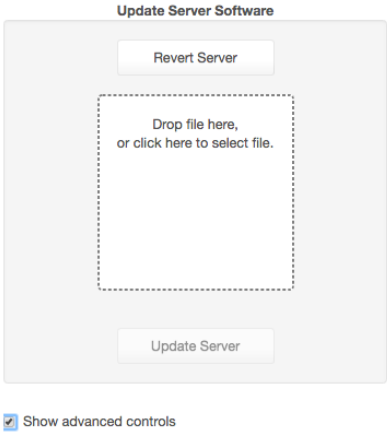
1. Make sure the Management Platform is powered and is working correctly.

2. Download the latest software from the ZeeVee website. Make note of the location of where the software was downloaded.

3. Login to the ZyPer Management Platform. Refer to Accessing ZyPer Management Platform (page 13) for more information.

4. Click the Server option at the left of the page.

5. Scroll down within the Server pain until you see the option to Update Server Software. Drag the latest software into the box and press Update Server to begin process. (Note: You can also revert the server to the previously installed version of software by clicking the Revert Server button) "Show advanced controls" must be enabled to use this option.



Important Notes:

The ZyPer MP update file will be available in five, platform-specific versions. Please use the correct version for the hardware platform being updated.

ZyPerMP Single NIC NUC computer: update_nuc_4.1.xxxxx.zyper
ZyPerMP Dual NIC NUC computer: update_nuc2004_4.1.xxxxx.zyper
ZyPerMP Proserver: update_proserver_4.1.xxxxx.zyper
ZyPerMP VMware: update_vm_4.1.xxxxx.zyper
ZyPerUHD60-EMP: update_zemp_4.1.xxxxx.zyper

- First generation ZMP NUC devices are not supported with the 3.0 release of ZMP API. These devices are running an incompatible version of the Linux Operating System and were last shipped by ZeeVee back in 2017. These units can be easily identified as they have the brand name "GigaByte" written on the underside of the unit.

- Customers using this older NUC that wish to upgrade to the 3.0 or newer ZMP API release should contact the ZeeVee sales team (zv_sales@kramerav.com) to purchase an updated ZMP Hardware device.

Redundancy Configuration Instructions

To configure redundancy, follow the steps below. The secondary server must be running for the redundancy fields to be visible in ZMP or the API.

Configuring redundancy through the API

Configuring the IP Address

1) Login to the main ZMP, or Master through telnet/ssh.

2) Issue the "set server redundancy all-servers" command to configure redundancy

IE: set server redundancy all-servers virtual-ip address 172.16.5.239 network-interface video

3) Use the "show server redundancy" command to review the redundancy configuration and confirm the changes

4) Login to the Secondary server, or Slave,through telnet.

5) Use the "show server redundancy" command to review the redundancy configuration and confirm the changes

Configure the preferred roles

1) Login to the Master ZMP through telnet/ssh.

2) Issue the "set server redundancy this-server" command to set the preferred master and slave states on the server.

IE: set server redundancy this-server preferred-master true preferred-slave false

3) Use the "show server redundancy" command to review the redundancy configuration and confirm the changes

4) Login to the Slave ZMP through telnet.

5) Use the "show server redundancy" command to review the redundancy configuration and confirm the changes

Configuring redundancy through ZMP

1) Login through you Master ZMP GUI with Chrome.

2) Open the Server Panel

3) Scroll down to the Redundancy fields

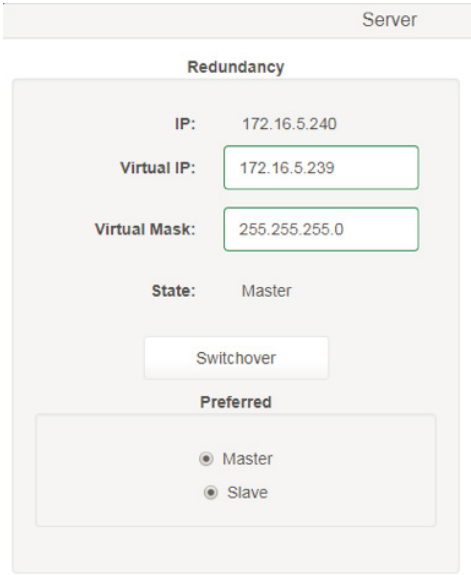
4) Set the fields listed below.

Virtual IP: The IP address that the Master and Slave servers will use. This IP address must be unique and available on the network as it will be used for telnet access for the API as well as ZMP.

Virtual Mask: The subnet mask for the virtual interface, must be correct for the IP address listed above and not it should not conflict with the main eth0 interface.

Preferred Roles Radio Button: The preferred roles for the server. This field is used to decide the Master or Slave upon both servers initializing at the same time. Although rare, this can occur.

State: The current role of the current Server connected to.



After configuration is complete on the Master, the information should populate to the Slave server. The preferred roles for the Slave server will still need to be configured. This can be done by logging into ZMP using the Slave server IP address and modifying the Preferred roles.

The "State" field will reflect the servers current state.

5) After the configuration changes are made, login into ZMP with the Virtual IP address configured above.

The server panel should show the correct redundancy information.

Note: The "switchover" button above will allow the servers to swap roles as needed.

Virtual interface on the ZMP.

Below is an example of the output of the "ifconfig" from the ZMP showing the virtual IP configured on the current master server.

```
eth0      Link encap:Ethernet  HWaddr 40:8d:5c:32:46:0e
          inet addr:172.16.5.240  Bcast:172.16.5.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX bytes:36015816 (36.0 MB)  TX bytes:31515642 (31.5 MB)

eth0:ZMP  Link encap:Ethernet  HWaddr 40:8d:5c:32:46:0e
          inet addr:172.16.5.239  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

eth0:avahi Link encap:Ethernet  HWaddr 40:8d:5c:32:46:0e
          inet addr:169.254.4.58  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX bytes:4873342 (4.8 MB)  TX bytes:4873342 (4.8 MB)
```

Important Note Regarding the Virtual Interface on the ZMP

If the Master ZMP goes offline the Slave ZMP will take over and become the new Master. There is a brief period of between 10 and 20 seconds where the Virtial IP address may not be availalbe on the network. This is due to the ARP Age Time parameter setting of the associated network.

Please consult your network switch documentation regarding this parameter and set it to the lowest setting possible.

Hardware Specifications (Older Intel NUC version)

CPU	• Intel® Pentium® Processor J5005
Operating System	• Linux Ubuntu 16.04
Internal Storage	• 64 GB SSD
Graphics	• Intel® HD Graphics 600
LAN	• Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 19V DC, 3.42 A
I/O	• 2 x HDMI 2.0a • 4 x USB 3.0, Type- A, female • 1 x RJ45 • 1 x 19V DC • 1 x Kensington lock slot • 2 x 3.5mm headset jacks (Not used)
Operating Temperature	• 0 °C to +40 °C
Storage Temperature	• -20 °C to +60 °C
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 4.55 in x 2.01 in x 4.57 in (115 mm x 51 mm x 111 mm)

Hardware Specifications (NUC-E version Feb 2022 to Mid 2025)

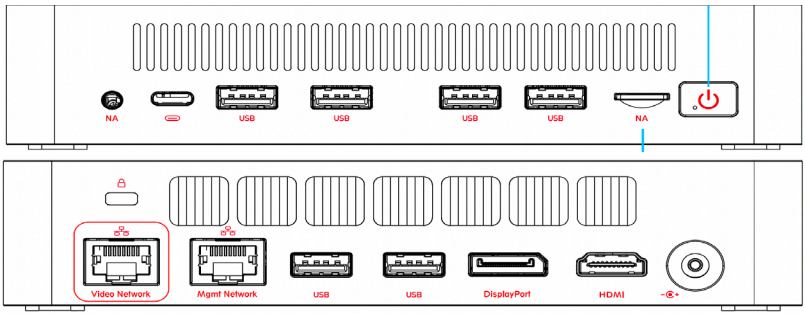
CPU	• Intel® Celeron® Processor N3350
Operating System	• Linux Ubuntu 20.04
Internal Storage	• 64 GB SSD
Graphics	• Intel® HD Graphics 500
LAN	• 1x 100mb LAN (Eth0), 1x Gigabit LAN (Eth1)
Internal Memory	• 4 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 19V DC, 3.42 A
I/O	• 2 x HDMI 2.0a • 3 x USB 3.0, Type-A, female • 2 x USB 2.0, Type-A, female • 2 x RJ45 • 1 x 19V DC • 1 x Kensington lock slot • 1 x 3.5mm headset jacks (Not used)
Operating Temperature	• 0 ºC to +40 ºC
Storage Temperature	• -20 ºC to +70 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 6.06 in x 1.25 in x 4.25 in (154 mm x 32 mm x 108 mm)

RoHS

Ethernet Port 0 = Video Port. DHCP default IP Address (Side with USB only)
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port. (Side with HDMI ports)
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

Hardware Specifications (NUC-F version Q4 2025 and beyond)

CPU	• Intel® Alder-N Lake N100
Operating System	• Linux Ubuntu 22.04
Internal Storage	• 128 GB SSD
Graphics	• Intel® UHD Graphics
LAN	• Dual Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 12V DC, 3A
Power Use	• 24W Peak, 18W Normal operation
I/O CE RoHS FC	• 1 x HDMI 2.0a • 1 x DisplayPort • 4 x USB 3.0, Type-A, female • 2 x USB 2.0, Type-A, female • 2 x RJ45 • 1 x 12V DC • 1 x Kensington lock slot • 1 x 3.5mm headset jack (Not used) • 1 x MicroSD Card Slot (Not used)
Operating Temperature	• 0 ºC to +45 ºC
Storage Temperature	• -20 ºC to +70 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 6.85 in x 1.42 in x 5.04 in (174mm x 36 mm x 128 mm)



Ethernet Port 0 = Video Port. DHCP default IP Address
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port.
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

Important Note: The Power/Reset button will reset the unit when button is released. Not when it is pressed. Do not press and hold the button in. Just give a quick press.

Hardware Specifications (Enterprise Grade Rack Mount)

CPU	• Intel® Xeon E3-1200 v5
Operating System	• Linux Ubuntu 16.04
Internal Storage	• 64 GB SSD
Graphics	• ASPEED AST2400 BMC
LAN	• Dual Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• 200W Low-Noise AC-DC power supply. Fan speed dynamically adjusts for load and environment.
I/O	• 1 x VGA (15-pin D-sub) • 2 x USB 2.0, Type-A, female • 2 x RJ45 (LAN) (Video Network and Management Network) • 1 x RS232 (9-pin D-sub)
Operating Temperature	• +10 °C to +35 °C
Storage Temperature	• -40 °C to +70 °C
Dimensions (W x H x D)	• 17.2 in x 1.7 in x 11.3 in • (437 mm x 43 mm x 287 mm)
Ambient Noise	• Measurement point was 1M distant, straight in front of unit • 43.5 db(A) - Startup and peak load condition • 32.8 - 34.4 db(A) - Expected range during typical load • 31.7 db(A) - Idle and very lightly loaded or cooler ambient conditions
Weight	• 8.45 lbs, (3.83 kg)

RoHS

Ethernet Port 0 = Video Port. DHCP default IP Address
Video Port connected to same network with ZyPer Endpoints
Ethernet Port 1 = Management Port.
Management Port connected to other network (if used)
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

Hardware Specifications (VMware Virtual Machine)

CPU	• Dual CPU's
Server Platform	• VMware ESXi 6.0 or later
Internal Storage	• 64 GB or greater
LAN	• Gigabit LAN
Internal Memory	• 8 GB or greater

North America: +1 (347) 851-7364 • ZV_sales_team@kramerav.com
International: +1 (347) 851-7364 • ZV_sales_team@kramerav.com
EMEA: +44 1494 956677 • ZV_sales_team@kramerav.com
DACH: +49 171 3620083 • ZV_sales_team@kramerav.com