

# ZyPer

## MultiCodec Management Platform

---

### User Manual

Updated October, 2022

API Release 2.5x



# 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)

support@zeevee.com

## Sales

### **North America:**

+1 (347) 851-7364 Phone

sales@zeevee.com

### **EMEA:**

+44 1494 956677 Phone

EMEAsales@zeevee.com



# Features and Package Contents

---

## 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 or ZyPerHD 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 / ZyPerHD 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 / ZyPerHD 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 15x15 array.
- Create and manage Multi-view displays with up to 19 sources. (ZyPer4K only)

# New in Release 2.3 / 2.4 / 2.5

---

## 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 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 in Release 2.3

---

### 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 in Release 2.3 / 2.5

---

### New/Updated Commands 2.3 (See API Command Listing)

- clone multiview
- dumpusb
- load encoderEdid
- set decoder osdStatusMode
- set device
- set multiview
- set preset

### New/Updated Commands 2.5 (See API Command Listing)

- load idleImage
- set decoder hdmi5vControl (ZyPer4K-XS and ZyPer4K-XR only)
- set device irProcessing (ZyPer4K only)
- set server discoverMode
- set terminal output

## Removed in Release 2.3

---

### The following items/features have been removed

- Mobile Support. The ZMP no longer supports mobile platforms such as tablets. (Note: this support was removed in release 2.2)
- **Note:** If non-tablet device somehow gets put into “Tablet” mode you can force it out of tablet mode by appending ?desktop=true to the end of the browser address. Example: <http://172.16.6.111/maestroz/app/?desktop=true>
- **Note:** Release 2.3.x will be the final release to support the ZyPerHD

# 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.....	7
Network Configuration.....	8
Fixed Mapping using DHCP.....	8
Static IP Configuration .....	9

## 2 Basic Operation

Accessing ZyPer Management Platform .....	11
Home Page .....	12
Device Status Indicators .....	13
Displaying Devices By Status.....	15
Displaying Devices By Name .....	16
Joining Encoders to Decoders.....	17
Unlinking Encoders and Decoders .....	18
Video Preview Stream .....	20
Source/Display Config Page .....	21
Configuring Encoders and Decoders .....	21
Deleting, Rebooting or Resetting an Encoder or Decoder.....	28
Sending a CEC Command .....	30
Sending a RS232 String Command.....	32
Updating Firmware.....	34
Creating Video Walls.....	35
Bezel Compensation.....	39
Creating a Multiview Screen .....	41
Creating Zones.....	45
Presets and Scheduling.....	48
Source Grid Page .....	53
Display Grid Page .....	58
Server Functions Page .....	63
Server Information .....	63
Setting the EDID Audio Mode.....	64
Server Reboot, Restart, and Shutdown .....	65
Help Page.....	67
Users and Roles Pages.....	68
Adding a New User .....	70
Adding or Editing a New Role.....	71

---

<b>3</b>	<b>Advanced Operation</b>	
	Accessing the API .....	75
	Using Telnet.....	75
	Getting Help .....	75
	Setting the Time Zone .....	78
	EDID Management .....	79
	Auto EDID Mode.....	79
	Using Custom EDID Data.....	79
	Using AJAX/JSON .....	81
	Login Authentication.....	81
	Command Request / Response.....	82
	Fast-Switched vs. Genlocked Mode .....	84
	API Command Listing .....	85
<b>4</b>	<b>Event Mechanism</b>	
	ZMP Event Mechanism .....	246
	Telnet Event Session.....	246
	Browser WebSocket .....	247
	API Polling .....	248
	Event List.....	248
<b>5</b>	<b>SNMP and LLDP Support</b>	
	SNMP Agent SNMPv3 Account Configuration.....	254
	SNMP Agent SNMPv2 Account Configuration.....	255
	SNMP Client.....	255
	SNMP Notifications.....	256
	More Technical Information on Linux Net-SNMP Package .....	257
<b>6</b>	<b>Appendix</b>	
	Updating the Software.....	259
	Using Mac OS X.....	260
	Using Windows.....	260
	Using ZyPer Management Platform .....	260
	Redundancy Configuration Instructions .....	264
	Hardware Specifications .....	267

1

# Getting Started

**zeevee**  
Signal Intelligence

**ZyPerMP**  
ZyPer Management Platform

Intelligent Video Distribution  
for Professionals





## Hardware Specifications (Intel NUC version)

CPU	<ul style="list-style-type: none"> <li>Intel® Pentium® Processor J5005</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 16.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>Intel® HD Graphics 600</li> </ul>
LAN	<ul style="list-style-type: none"> <li>Gigabit LAN</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>8 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>Input: 100 ~ 240 V AC</li> <li>Output: 19V DC, 3.42 A</li> </ul>
I/O	<ul style="list-style-type: none"> <li>2 x HDMI 2.0a</li> <li>4 x USB 3.0, Type- A, female</li> <li>1 x RJ45</li> <li>1 x 19V DC</li> <li>1 x Kensington lock slot</li> <li>2 x 3.5mm headset jacks (Not used)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>0 °C to +40 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-20 °C to +60 °C</li> </ul>
VESA	<ul style="list-style-type: none"> <li>VESA Bracket included</li> <li>Supports 75 x 75 and 100 x 100 mm</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>4.55 in x 2.01 in x 4.57 in (115 mm x 51 mm x 111 mm)</li> </ul>



RoHS



## Hardware Specifications (NUC version Feb 2022 and beyond)

CPU	<ul style="list-style-type: none"> <li>Intel® Celeron® Processor N3350</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 20.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>Intel® HD Graphics 500</li> </ul>
LAN	<ul style="list-style-type: none"> <li>1x 100mb LAN (Eth0), 1x Gigabit LAN (Eth1)</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>4 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>Input: 100 ~ 240 V AC</li> <li>Output: 19V DC, 3.42 A</li> </ul>
I/O	<ul style="list-style-type: none"> <li>2 x HDMI 2.0a</li> <li>3 x USB 3.0, Type-A, female</li> <li>2 x USB 2.0, Type-A, female</li> <li>2 x RJ45</li> <li>1 x 19V DC</li> <li>1 x Kensington lock slot</li> <li>1 x 3.5mm headset jacks (Not used)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>0 °C to +40 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-20 °C to +70 °C</li> </ul>
VESA	<ul style="list-style-type: none"> <li>VESA Bracket included</li> <li>Supports 75 x 75 and 100 x 100 mm</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>6.06 in x 1.25 in x 4.25 in (154 mm x 32 mm x 108 mm)</li> </ul>



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 (Enterprise Grade Rack Mount)

CPU	<ul style="list-style-type: none"> <li>Intel® Xeon E3-1200 v5</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 16.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>ASPEED AST2400 BMC</li> </ul>
LAN	<ul style="list-style-type: none"> <li>Dual Gigabit LAN</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>8 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>200W Low-Noise AC-DC power supply. Fan speed dynamically adjusts for load and environment.</li> </ul>
I/O	<ul style="list-style-type: none"> <li>1 x VGA (15-pin D-sub)</li> <li>2 x USB 2.0, Type-A, female</li> <li>2 x RJ45 (LAN) (Video Network and Management Network)</li> <li>1 x RS232 (9-pin D-sub)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>+10 °C to +35 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-40 °C to +70 °C</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>17.2 in x 1.7 in x 11.3 in</li> <li>(437 mm x 43 mm x 287 mm)</li> </ul>
Ambient Noise	<ul style="list-style-type: none"> <li>Measurement point was 1M distant, straight in front of unit</li> <li>43.5 db(A) - Startup and peak load condition</li> <li>32.8 - 34.4 db(A) - Expected range during typical load</li> <li>31.7 db(A) - Idle and very lightly loaded or cooler ambient conditions</li> </ul>
Weight	<ul style="list-style-type: none"> <li>8.45 lbs, (3.83 kg)</li> </ul>



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

Power Button



Status LEDs

---

## 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 8).

### 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®

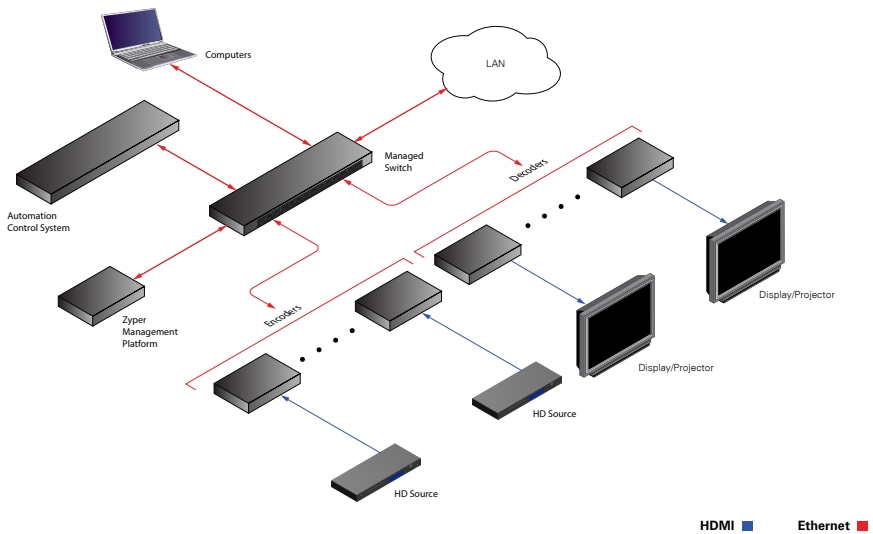
- a. Connect an Ethernet cable from your computer to the same switch as above.
- b. Open the Terminal application.

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 \(page 7\)](#) for more information.
7. Click the **Login** button.

## Sample Application Diagram



## 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 Management Platform

```
show server info
server(192.168.0.22);
  server.gen; hostname=zyper.local, version=2.3.36999,
previousVersion=2.2.36870,
macAddress=94:c6:91:a0:47:fc, serialNumber=ZZM1K400011D
  server.gen; uptime=3d:21h:25m:24s, freeMem= 6.71GB, bootCount=173
  server.gen; runningInVm=false
  server.ipActive; ipServerAddr=192.168.0.22, ipManagementAddr=NA,
gatewayAddr=none, dnsAddr=none
  server.time; time="Tue Aug 31 08:43:59 2021",
timezone=America/New_York
  server.pollStats; count=0, interval: 0-minutes, monListSize=0
  server.license; productID=F9188182-AF72-C6C8-92C6-94C691A047FC,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=6, allDevices=12,
allDevicesUp=6, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_0.0.0.0=1, num_2.0.4.0=2,
num_4.1.2.0=3
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(192.168.0.22);
  server.gen; hostname=zyper.local, version=2.3.36999,
previousVersion=2.2.36870,
macAddress=94:c6:91:a0:47:fc, serialNumber=ZZM1K400011D
  server.gen; uptime=3d:21h:25m:24s, freeMem= 6.71GB, bootCount=173
  server.gen; runningInVm=false
  server.ipActive; ipServerAddr=192.168.0.22, ipManagementAddr=NA,
gatewayAddr=none, dnsAddr=none
  server.time; time="Tue Aug 31 08:43:59 2021",
timezone=America/New_York
  server.pollStats; count=0, interval: 0-minutes, monListSize=0
  server.license; productID=F9188182-AF72-C6C8-92C6-94C691A047FC,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=6, allDevices=12,
allDevicesUp=6, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_0.0.0.0=1, num_2.0.4.0=2,
num_4.1.2.0=3
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.

```

restart server

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.*

Using the `set server ip` command can be used to set the IP Address of the Management Platform. Refer to [API Command Listing \(page 84\)](#) for a full listing of available commands.

```

set server ip server static 192.168.1.26 255.255.255.0 none none
reboot

Success

```



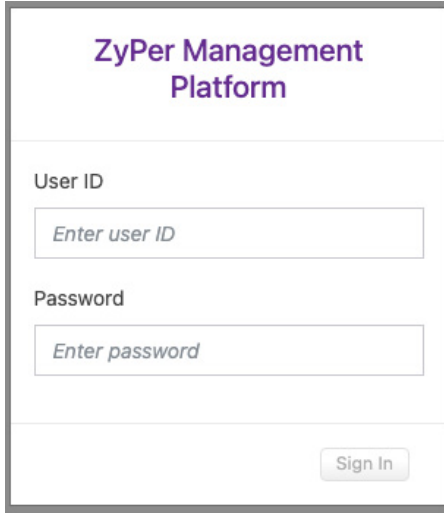
# 2

## 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.



The image shows a login screen for the ZyPer Management Platform. At the top, the text "ZyPer Management Platform" is displayed in a purple font. Below this, there are two input fields: "User ID" and "Password". The "User ID" field contains the placeholder text "Enter user ID" and the "Password" field contains the placeholder text "Enter password". At the bottom right of the form, there is a "Sign In" button.

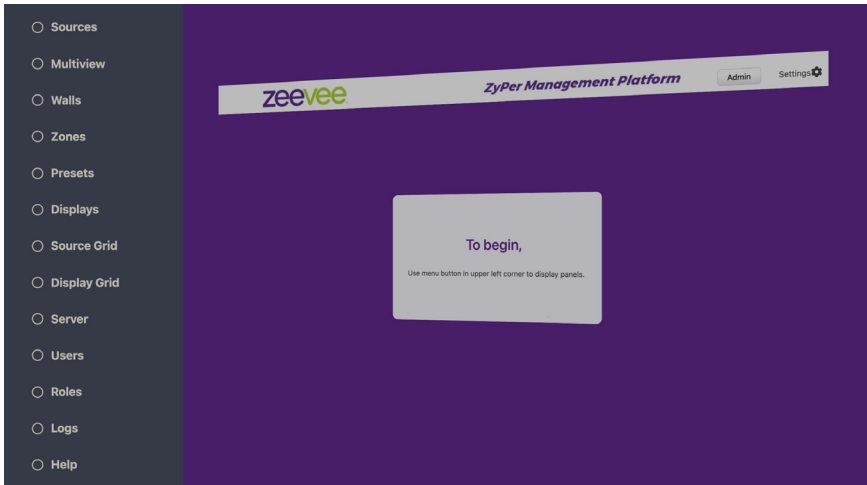
3. Enter the required information in the **Username** and **Password** fields. The default username and password is `admin`. The username and password are case-sensitive. The "admin" password may be changed by the user at any time.
4. Click the **Sign In** button.
5. The **Home** page will be displayed. 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 ZMP \(page 11\)](#) for more information.
2. The **Home** page will be displayed. The **Sources** pane displays all available *encoders* . The **Displays** pane displays available *decoders*. *Walls, Multiview, Zones, Presets, Source Grid, Display Grid, Server, Users* 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

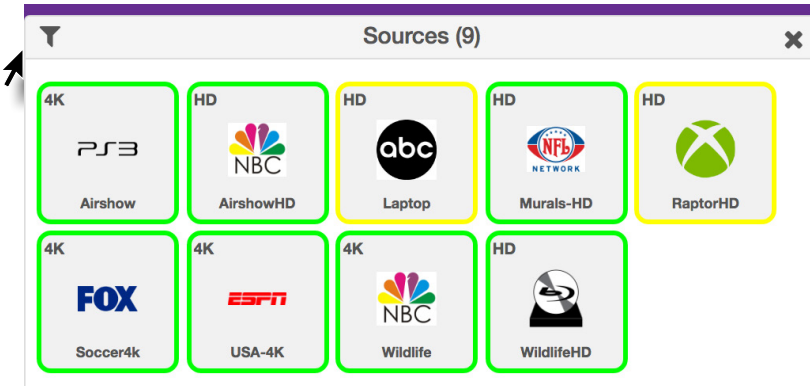


1. The Ethernet cable is *disconnected* between the switch and the encoder / decoder.
2. The encoder / decoder may not be powered.

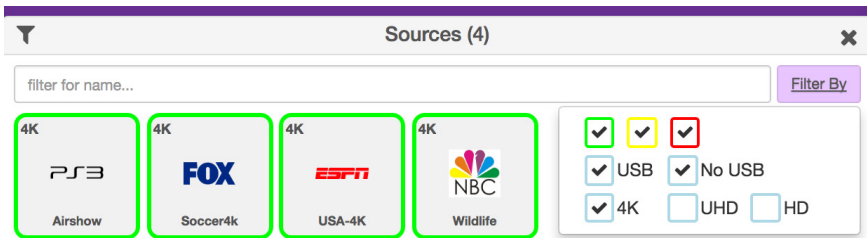
## Displaying Devices By Status

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, UHD 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. ZyPerHD and ZyPerUHD sources are filtered out.



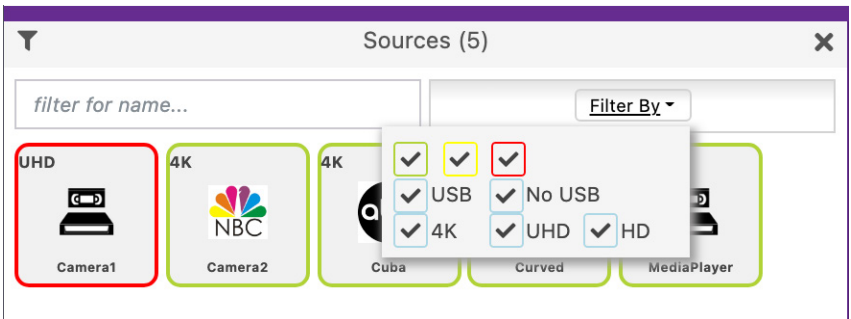
3. Select every box from the **Filter** to show all devices.

## Displaying Devices By Name

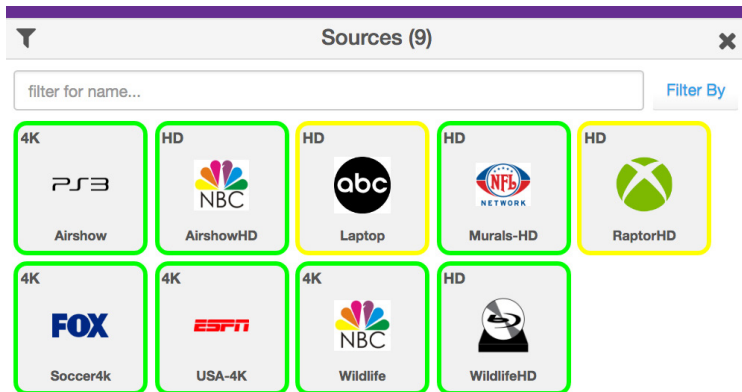
Both the **Sources** and **Displays** windows contain a **Filter by name** field. Use this field to enter the name of the desired device(s) to be displayed.

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 \(page 21\)](#).



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*.

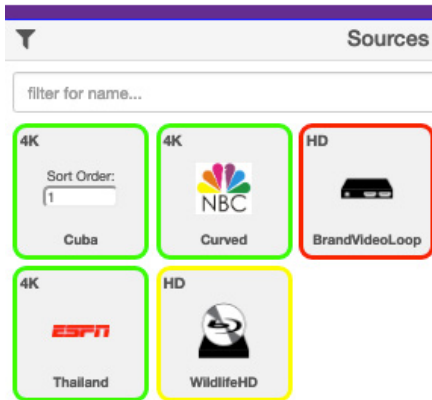
## Manually Setting Sort Order

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.



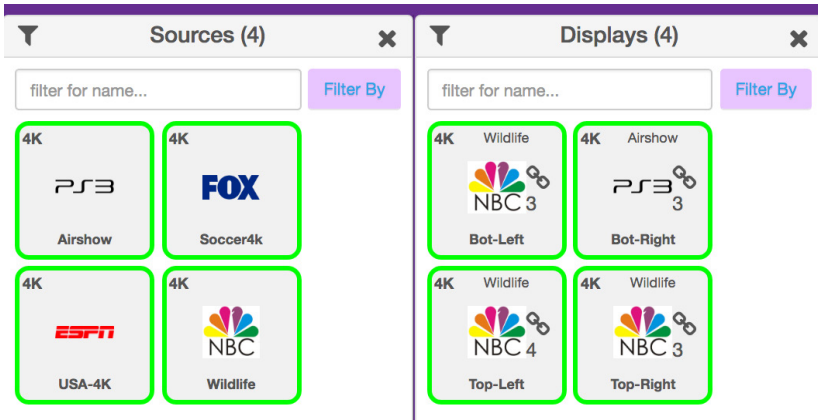
2. Sort Order numbering runs from top left to bottom right. In the image above assuming only 5 sources, the top three would be location 1, 2 and 3; while the bottom two would be locations 4 and 5.



## 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 \(page 21\)](#).

1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) 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.



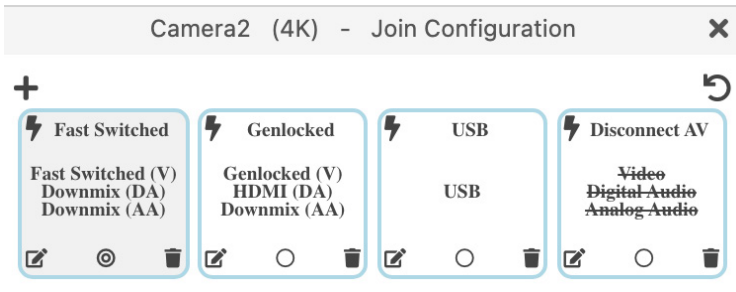
4. Continue the *join* process as desired. Note that joining an encoder with a decoder that is already joined, will replace the previous *join* operation.

The join command can also be used to perform the same operation. See [API Command Listing \(page 85\)](#) for more information.

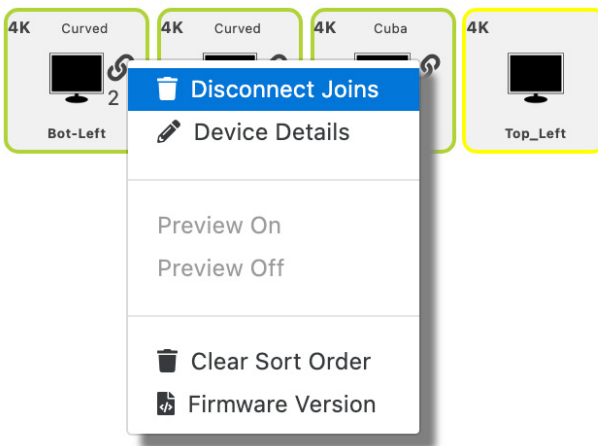
## Unlinking Encoders and Decoders

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 ZMP \(page 11\)](#) 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)

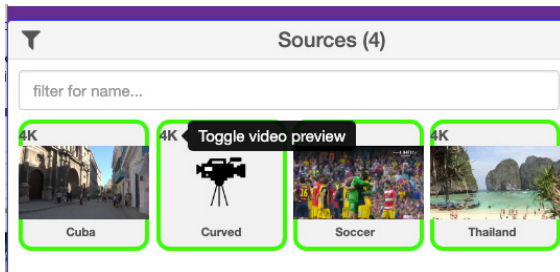


## Video Preview Stream

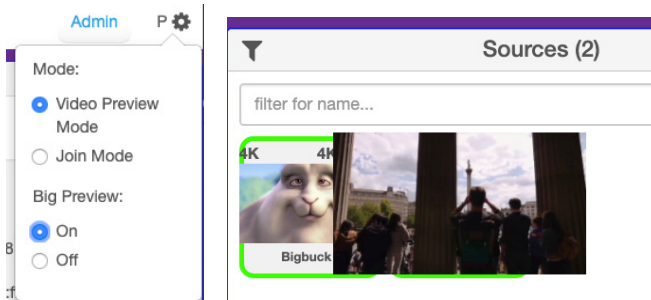
It is possible to view a small thumbnail preview of the active stream in the GUI within either the encoder or decoder box. Once preview streams are enabled, you can see them by following the steps below: (Supported on ZyPer4K and ZyPerUHD only) Not supported on ZyPer4K-XS or ZyPer4K-XR units.

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 ZMP \(page 11\)](#) 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

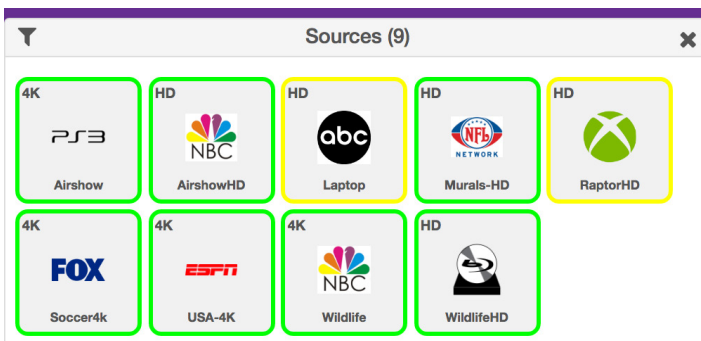
### Configuring Encoders and Decoders

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.

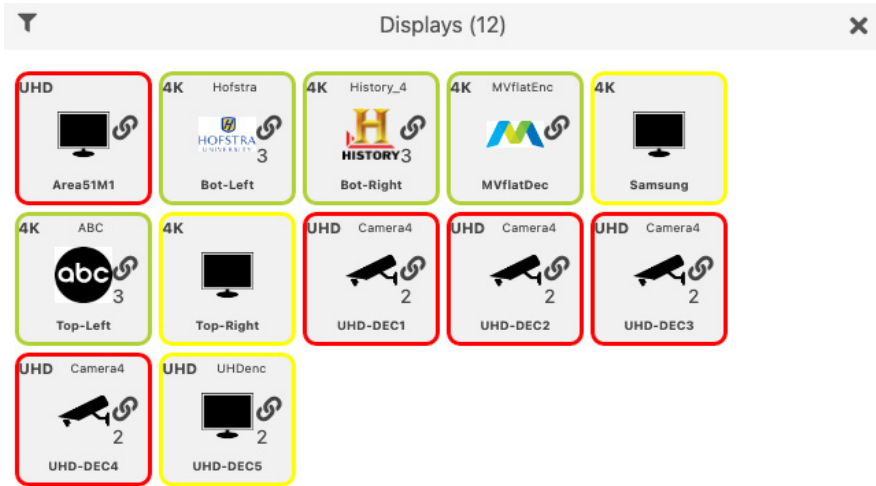
1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Source or Displays** tab at the left of the page. [Joining Encoders to Decoders \(page 18\)](#)



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)



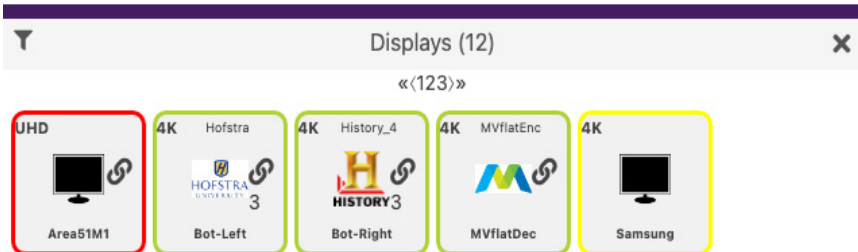
All available decoders will be displayed under the **Displays** window. (Maximum 25 per page. Deployments with more than 25 decoders will cycle thru pages)



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 browser address to set max Sources/Displays to 5. Valid numbers are 1 thru 25.

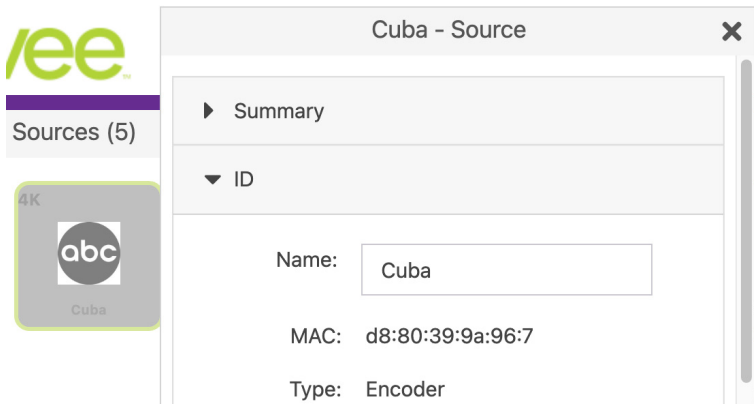
Example: <http://172.16.6.111/maestroz/app/?max=5>



- Left-click the desired encoder name to display the context menu. In the example below, we will select the encoder named "Cuba".
- A menu will appear with options for Summary, ID, Status, Config and Actions.



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



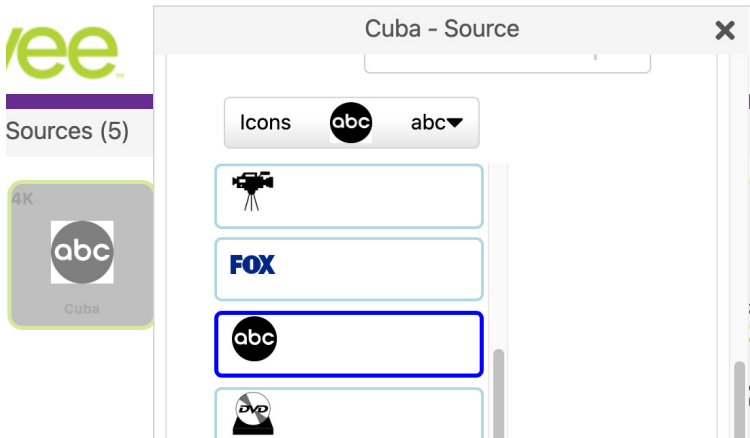
- Enter a description for the source in the **Name** field. By default, the Management Platform will use the MAC address of the encoder. This field cannot be blank. Names cannot contain spaces. In this example, we will use "ESPN1".

Naming a device can also be done using the `set device general name` command. See [API Command Listing \(page 85\)](#) for more information.

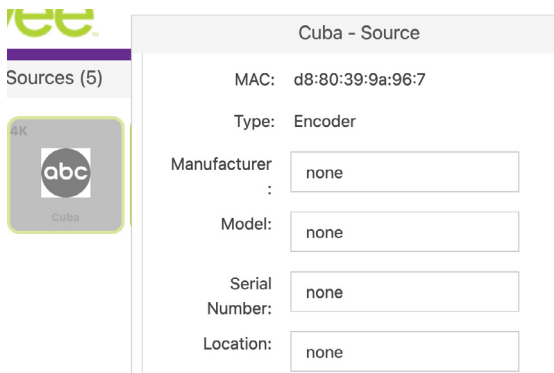
*Each encoder and decoder must have a unique name. Duplicate names are prohibited. It should be noted that assigning a name to an encoder does not actually replace the MAC address. The Name field provides a descriptive identifier, rather than a MAC address.*

- 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 \(page 85\)](#) for more information.



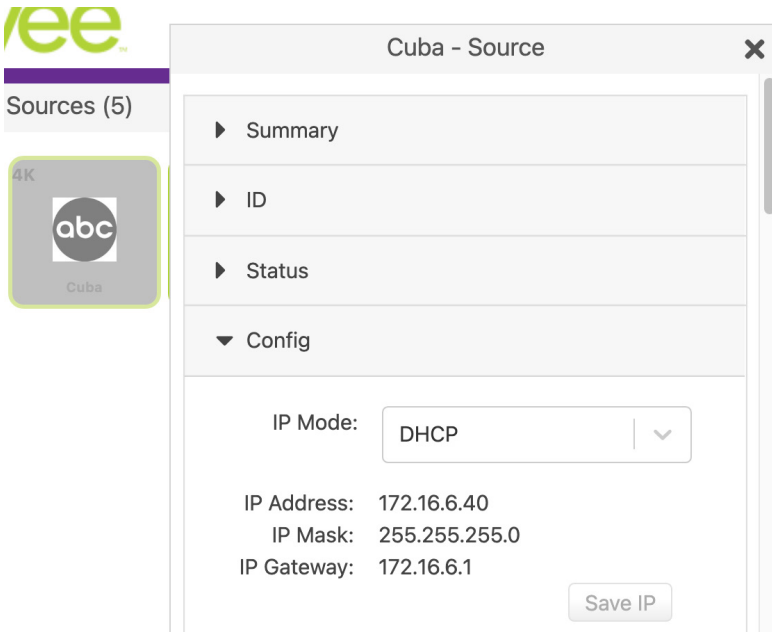
- Enter the **Manufacturer**, **Model**, and **Serial Number** of the source or display in the appropriate fields. By default, these fields are set to "none". These fields must not be blank. If a custom value is provided, it must *not* contain spaces.



The **Manufacturer**, **Model**, and **Serial Number** can also be assigned using the following commands, respectively. See [API Command Listing \(page 84\)](#) for more information.

- ▶ `set device source-display manufacturer`
- ▶ `set device source-display model`
- ▶ `set device source-display serialNumber`

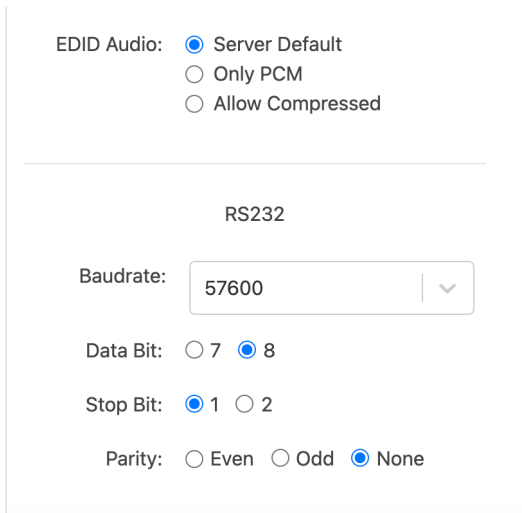
10. Enter the location of the source or display device in the **Location** field. By default, this field is set to “Unknown”. This field must not be blank. If a custom value is provided, it must *not* contain spaces.
11. The `set device source-display location` command can also be used to set these values. See [API Command Listing \(page 85\)](#) 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.



The screenshot shows the 'Cuba - Source' configuration window. On the left, there is a sidebar with the 'iVee' logo and a 'Sources (5)' section containing a source card for 'Cuba' with a '4K' label and an 'abc' logo. The main window has a title bar 'Cuba - Source' and a close button. The content is organized into sections: 'Summary', 'ID', 'Status', and 'Config'. The 'Config' section is expanded to show the IP Mode set to 'DHCP'. Below this, the IP Address is '172.16.6.40', the IP Mask is '255.255.255.0', and the IP Gateway is '172.16.6.1'. A 'Save IP' button is located at the bottom right of the configuration area.



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.



EDID Audio:  Server Default  
 Only PCM  
 Allow Compressed

---

RS232

Baudrate:  ▼

Data Bit:  7  8

Stop Bit:  1  2

Parity:  Even  Odd  None

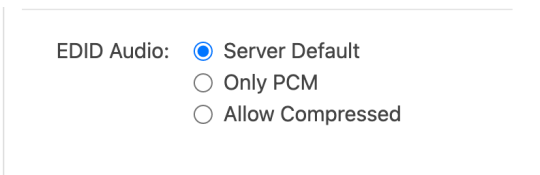
The **Device IP** settings can also be assigned using the `set device ip dhcp` and `set device ip static` commands. When assigning **RS232** settings from the command line, use the `set device rs232` command. See [API Command Listing \(page 85\)](#) for more information.

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.



EDID Audio:  Server Default  
 Only PCM  
 Allow 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 ZMP \(page 11\)](#) for more information.
2. Left-click the desired source name and select the **Status** option from the context menu.

▼ Status	
State:	Up
Input:	HDMI
Cable:	Connected
HDCP:	Inactive
Horizontal Resolution:	3840
Vertical Resolution:	2160
Refresh Rate:	30.000
Video Multicast Address:	224.1.1.14
HDMI Audio Multicast Address:	224.1.1.16
Analog Audio Multicast Address:	0.0.0.0
Firmware:	4.0.1.0

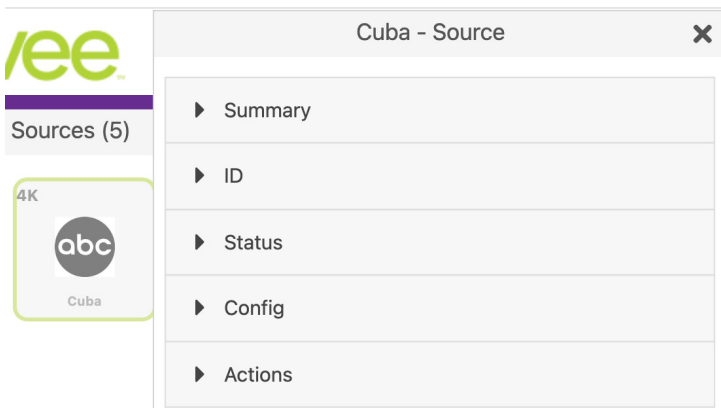
## Deleting, Rebooting or Resetting an Encode or Decoder

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 ZMP \(page 11\)](#) 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.



The screenshot shows a web interface with a dropdown menu titled "Actions". Inside the menu, there are two sections: "CEC" and "RS232". The "CEC" section contains "On" and "Off" buttons, and a "Hex:" label followed by a text input field. The "RS232" section contains a "Command:" label followed by a text input field. Below these sections are four buttons: "Reboot", "Factory Defaults", and "Delete".

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

### Are you sure?

Would you like to delete device Cuba?

Cancel

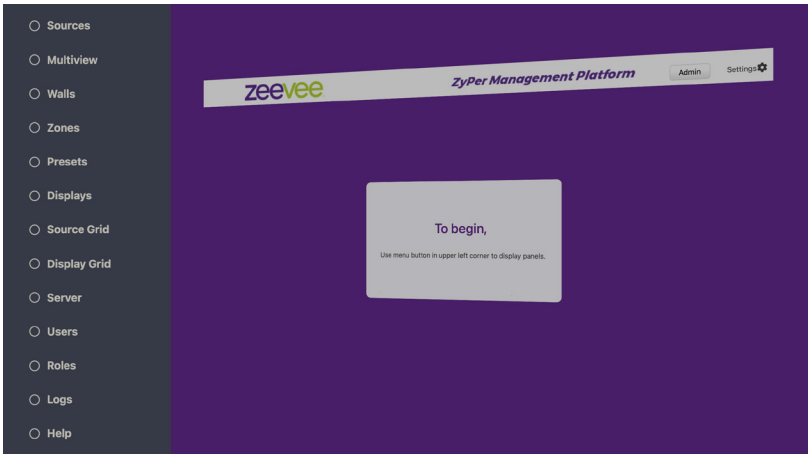
Delete

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.

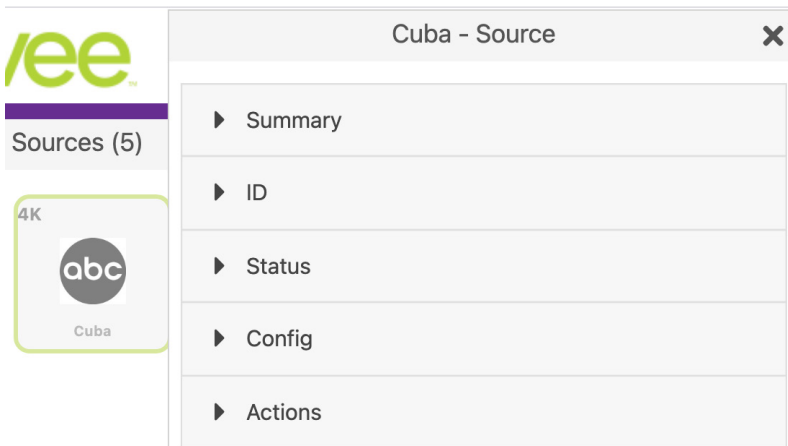
## Sending a CEC Command

Both the ZyPer4K and ZyPerUHD can send CEC on/off commands from within ZMP. The ZyPer4K can send additional CEC hex commands as well.

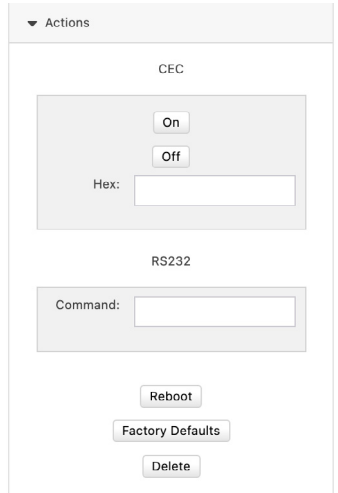
1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Source** tab at the left of the page.



- ▶ To send a CEC command:
  - a. Left-click on the desired encoder or decoder name and select **Actions** from the context menu.



- b. Click the desired **CEC On** or **OFF** button. (ZyPer4K or ZyPerUHD)



The screenshot shows a web interface titled "Actions" with a dropdown arrow. It contains two main sections: "CEC" and "RS232".

- CEC Section:** Contains two buttons labeled "On" and "Off". Below them is a text input field labeled "Hex:".
- RS232 Section:** Contains a text input field labeled "Command:".

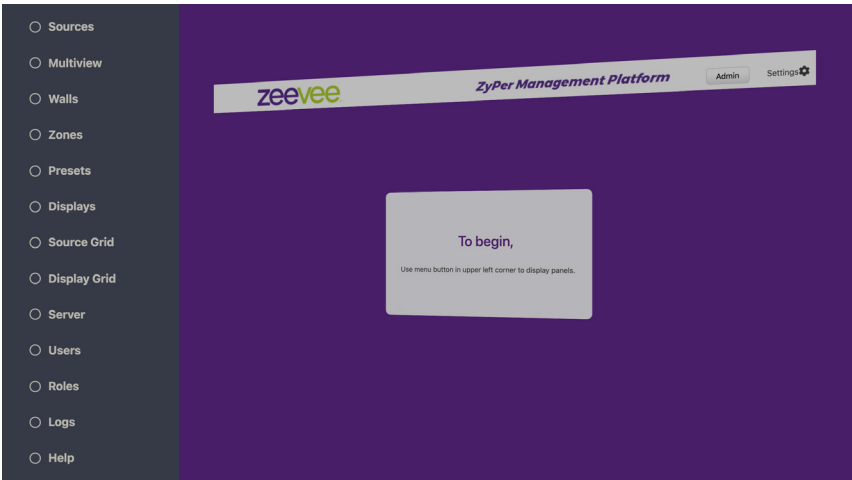
At the bottom of the interface, there are three buttons: "Reboot", "Factory Defaults", and "Delete".

3. The ZyPer4K 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.

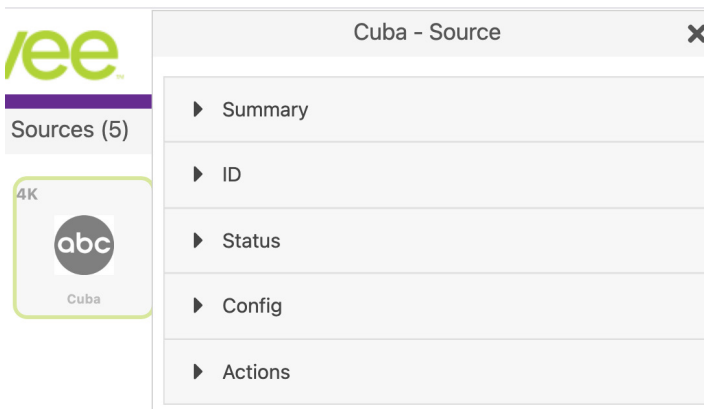
## Sending an RS232 String Command

Both the ZyPer4K, ZyPerUHD and ZyPerHD 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 ZMP \(page 11\)](#) for more information.
2. Click the **Source** 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.

The screenshot shows a web interface with a dropdown menu titled "Actions". Inside the menu, there are two main sections: "CEC" and "RS232".

The "CEC" section contains a sub-panel with two buttons labeled "On" and "Off", and a text input field labeled "Hex:".

The "RS232" section contains a text input field labeled "Command:".

Below the "RS232" section, there are three buttons: "Reboot", "Factory Defaults", and "Delete".



## Updating Firmware

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

1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Source** 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. Drag and drop the appropriate firmware update file into the box and click on the **“Update Device”** button.

▼ Actions

CEC On

CEC Off

Hex:

Reboot

Factory Defaults

Delete

**Update Device Firmware**

Drop file here,  
or click here to select file.

Update Device

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.

### Are you sure?

Would you like to update the device NBC with  
Z4K\_Firmware\_HDMI2.0\_v4\_0\_1\_0.apz?

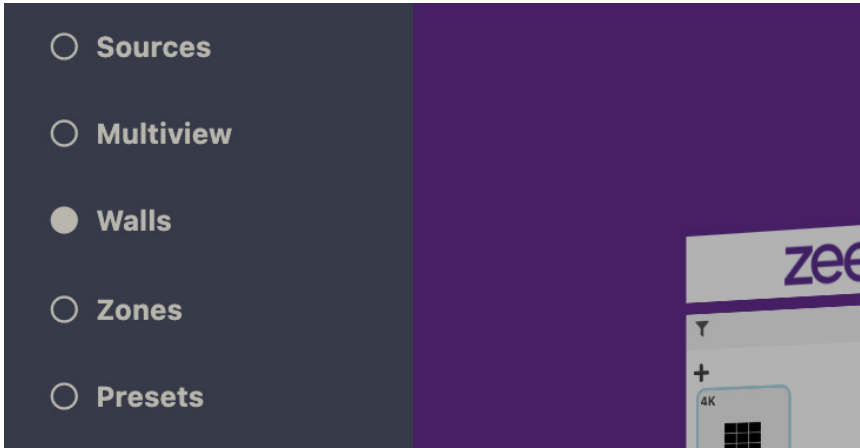
Cancel

Update

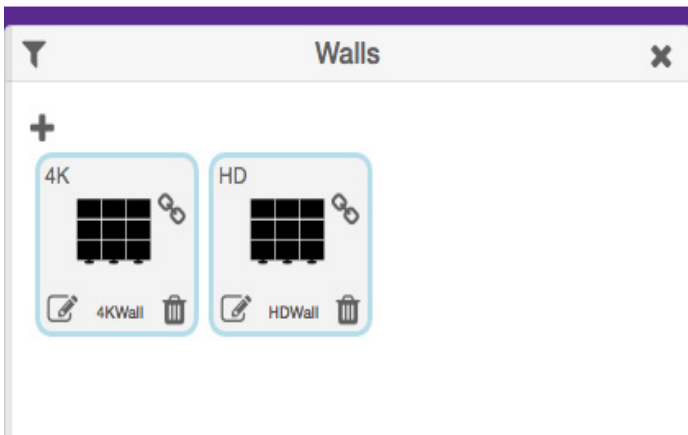
## Creating Video Walls

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.

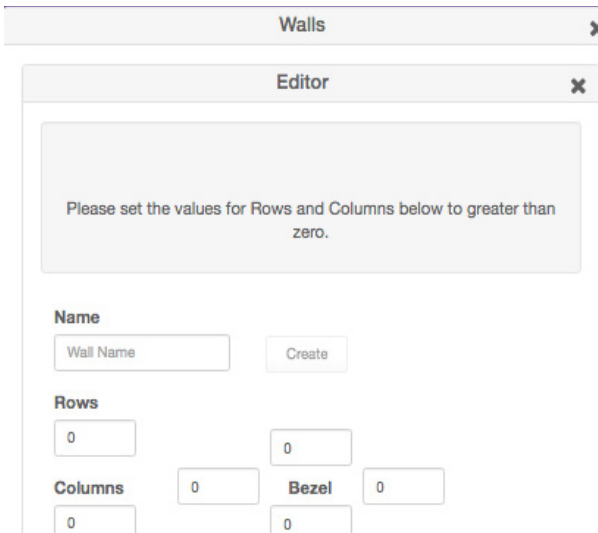
1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Walls** tab at the left of the page.



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



- This will bring up the wall **Editor**.

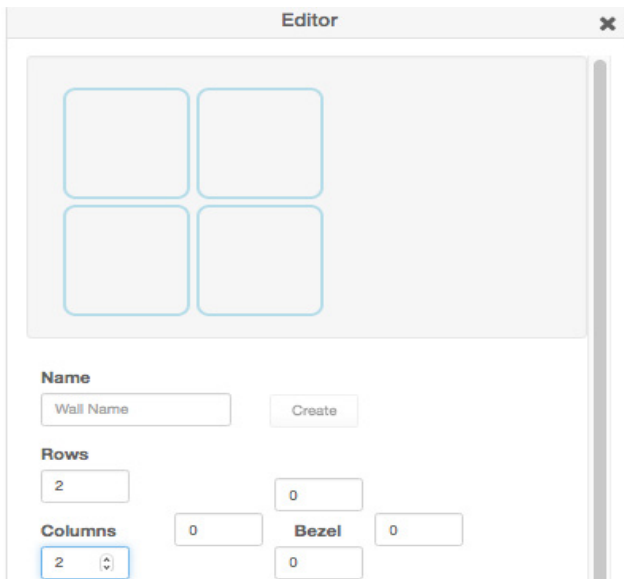


The screenshot shows a window titled "Walls" with a sub-window titled "Editor". Inside the "Editor" window, there is a message box that says "Please set the values for Rows and Columns below to greater than zero." Below the message box, there are several input fields and a button:

- Name:** A text input field labeled "Wall Name" and a "Create" button.
- Rows:** A text input field containing the value "0".
- Columns:** A text input field containing the value "0".
- Bezel:** A text input field containing the value "0".

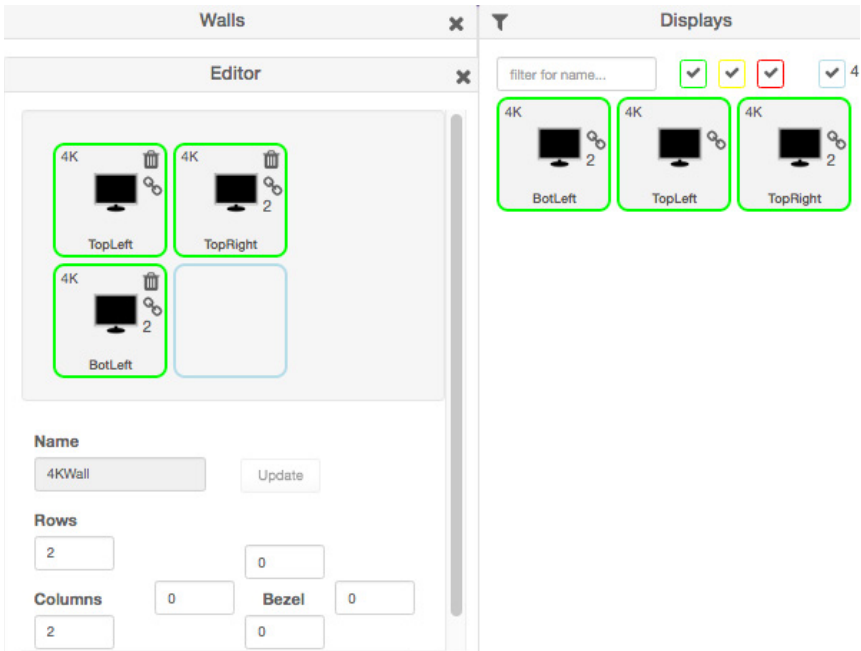
- Enter the number of display columns in the **Number of Columns** field.
- Enter the number of display rows in the **Number of Rows** field.

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



The screenshot shows the "Editor" window with a 2x2 grid of video wall displays. The "Rows" field is set to "2" and the "Columns" field is set to "2". The "Bezel" field is set to "0". The "Name" field is "Wall Name" and the "Create" button is visible.

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.

### Are you sure?

Would you like to remove "wall1" video wall?

Cancel

Remove

## 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

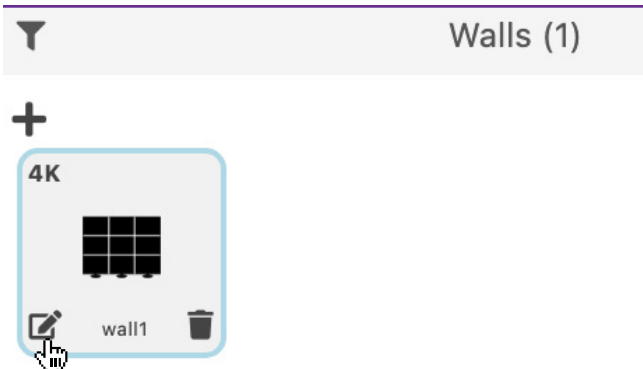


With Bezel Compensation

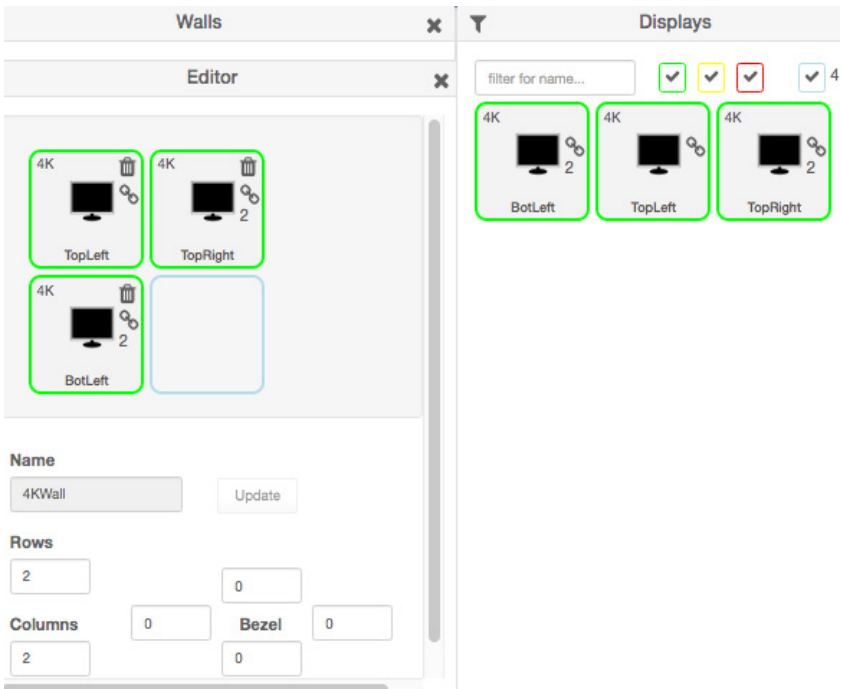
It is recommended that when the video wall is set up for the first time, that these values be set to zero. *Bezel compensation* can be changed at any time.

*Bezel compensation is always measured in pixels.*

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



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



- Enter the desired values, in pixels, for each bezel field: **Bezel Top**, **Bezel Bottom**, **Bezel Left**, **Bezel Right**.
- Once the desired values have been entered, click the **Update** button. This will save the new settings.
- Check the picture on the displays. Repeat steps 2 - 5 as necessary.
- Note:** Only the ZyPer4K allows bezel adjustment. Video walls with the ZyPerHD or ZyPerUHD do not allow bezel adjustment.
- 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 `/?desktop=true`. **Example:** `http://172.16.6.111/maestroz/app/?desktop=true`

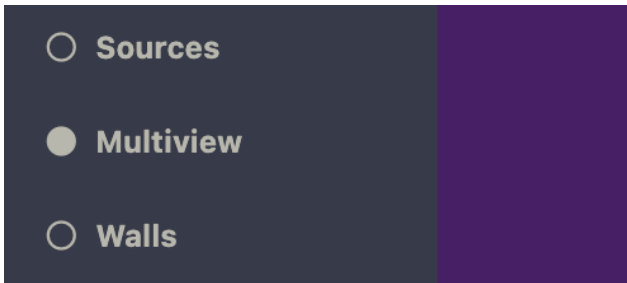
---

## Creating a Multiview Screen

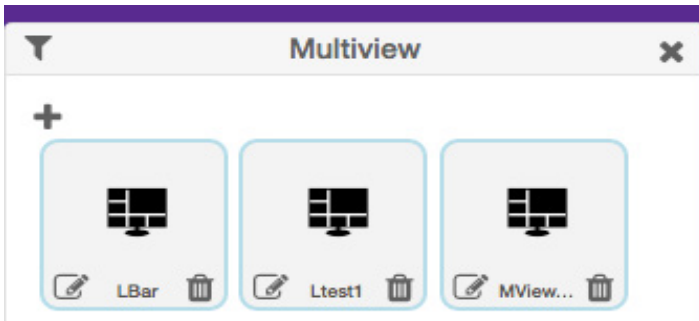
One of the purposes of the Management Platform is to create and manage multiview screens. A multiview screen is a collection of sources arranged on a single display. There can be up to 19 sources displayed on a single display in a variety of preset patterns.

**Note:** Only the ZyPer4K versions with the HDMI 2.0 feature support multiview screens. This feature is not supported by the ZyPerHD, ZyPerUHD or ZyPer4K units with only HDMI 1.4 support.

1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) 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.

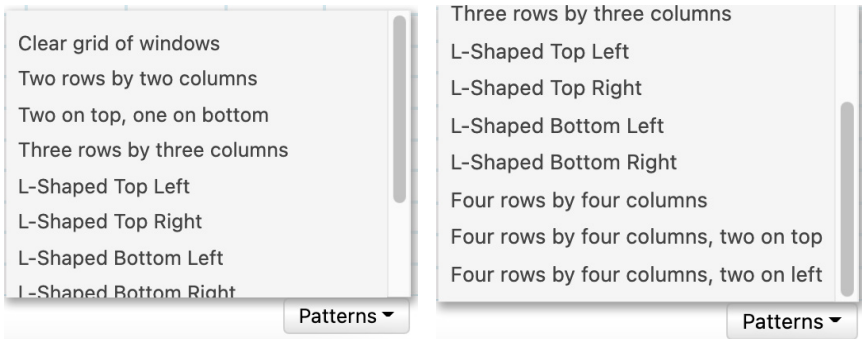


4. This will bring up the Multiview editor where you can create and customize a multiview display.

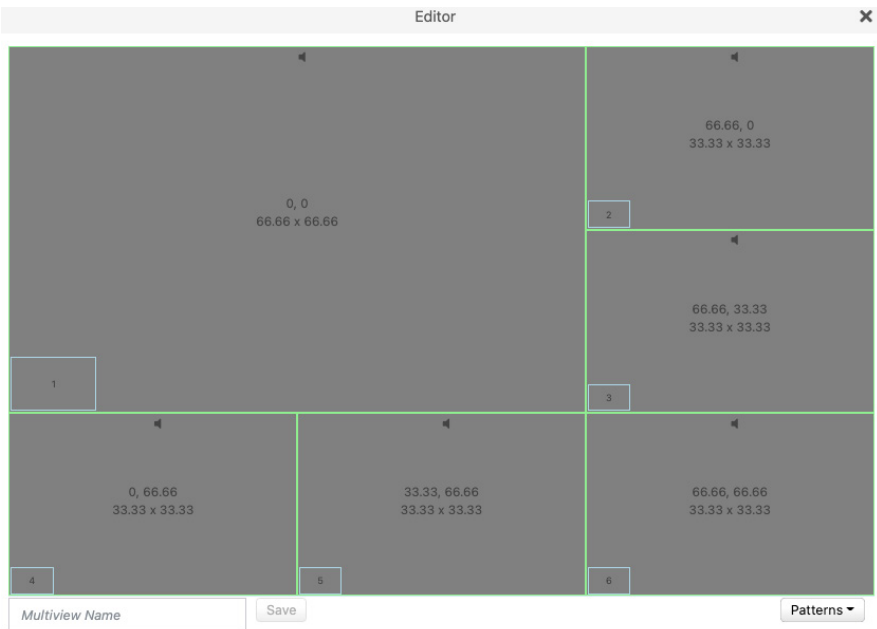




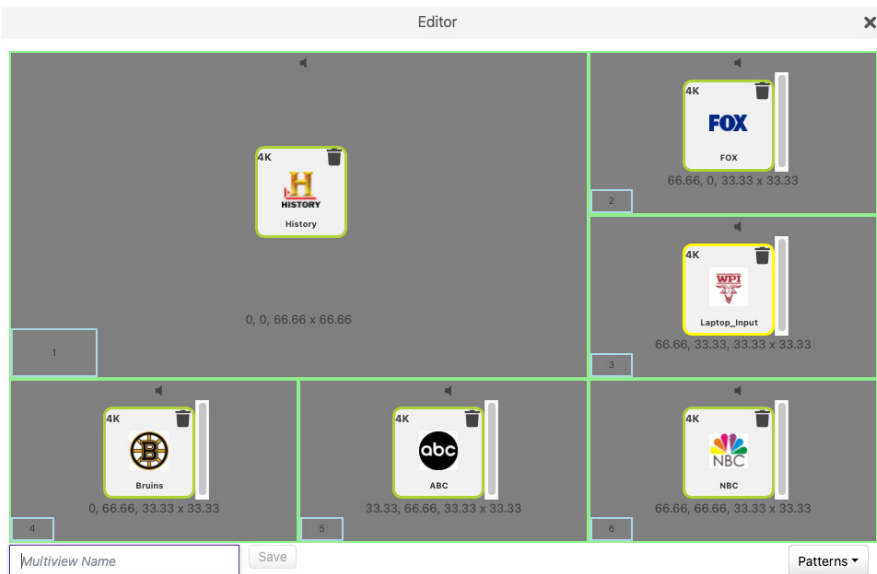
5. Click on the **Patterns** button in the lower right corner to bring up a list of pre-defined patterns.



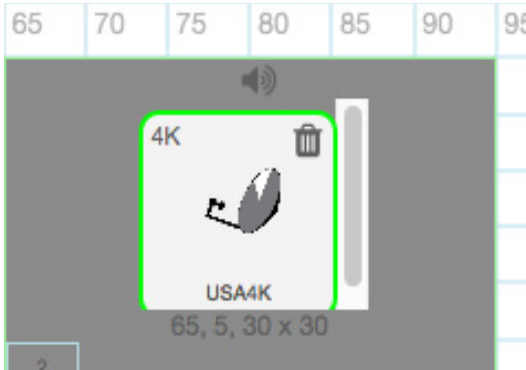
6. After selecting a pattern you will see the grid fill in with the places to drag sources.



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.

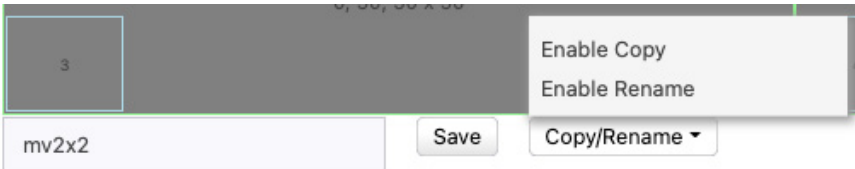


8. If desired, assign Audio from one of the Sources. This is done by clicking on one of the small Audio symbols in the top center of each Window. Note that only audio from one source is supported.



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.

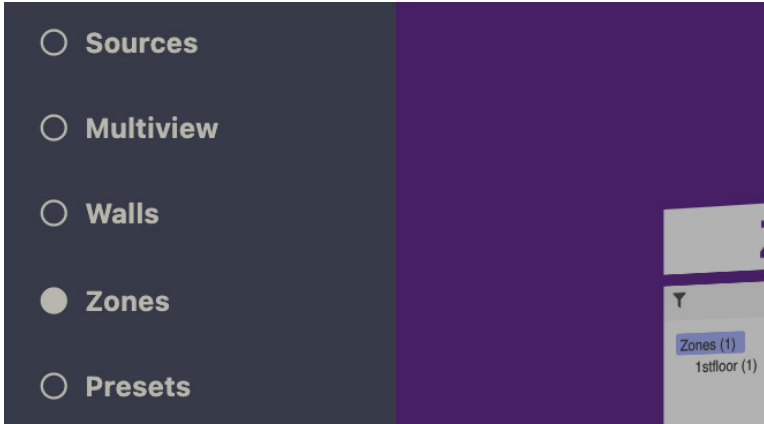
Click "Enable Rename", enter the new name for existing multiview and click Save.

---

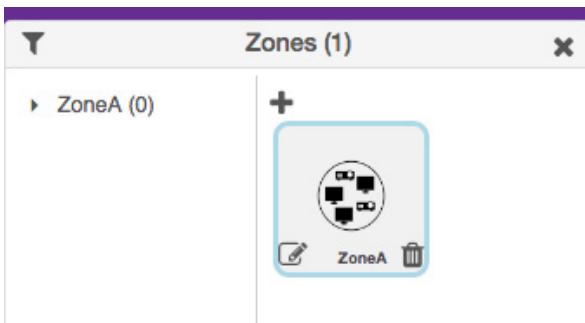
## Creating Zones

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*.

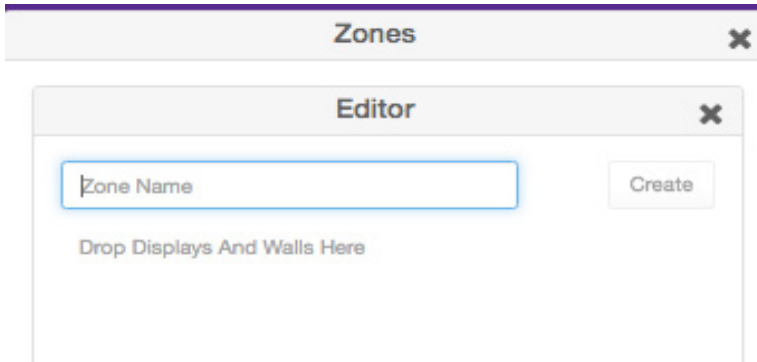
1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Zones** option at the left of the page.



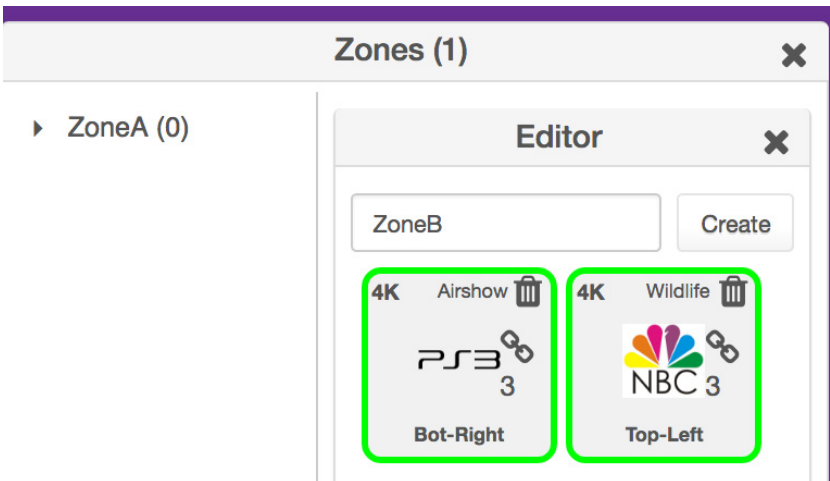
3. In the Zones window, click on the + symbol to add a Zone.



- This will open the Zones Editor. Drag Display and Video Walls into the Zone from the **Displays** or **Walls** windows.

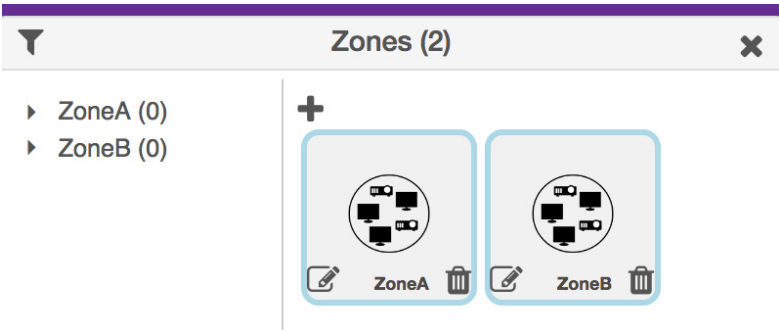


- Continue adding the desired displays (or *video walls*) to the drop-pane.



- 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*.

- Click the **Create** button to save the *Zone*. Close the editor window to exit without saving changes.
- The new *Zone* will appear under the **Zones** window.



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

#### Are you sure?

Would you like to delete zone "1stfloor"?

Cancel

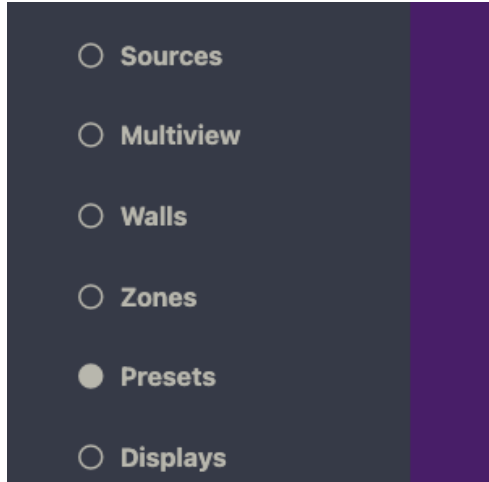
Delete

---

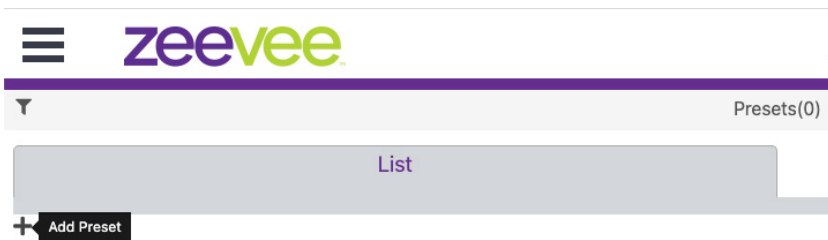
## Creating Presets and Schedules

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.

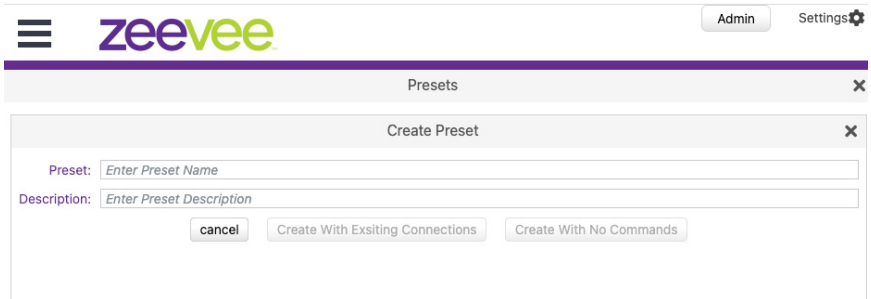
1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Presets** option at the left of the page.



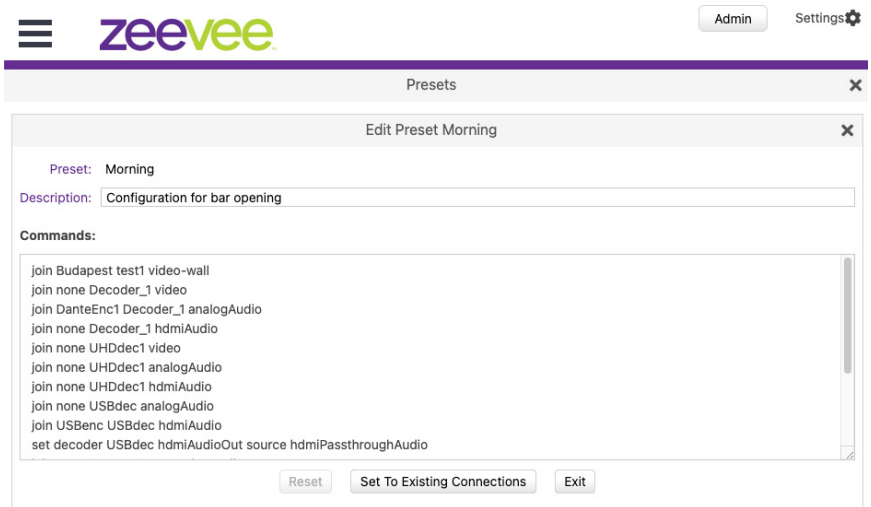
3. In the Presets/List window, click on the **+** symbol to add a Preset.



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



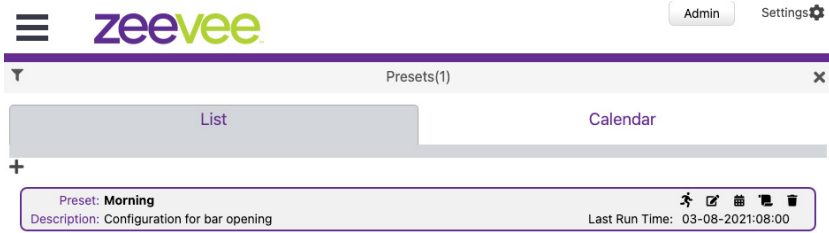
- You can now Create the preset using the current state of the system. "Create with Existing Connections" or you can "Create with No Commands" and add the API commands manually.



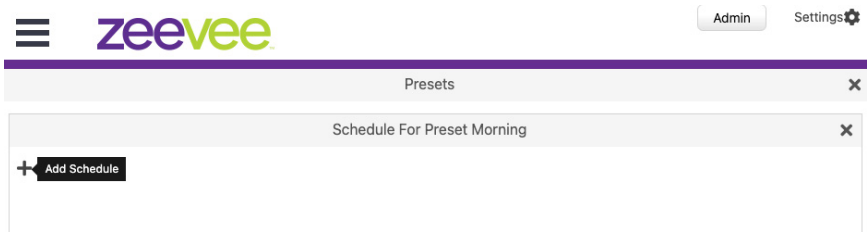
- The example above is "Create with Existing Connections". Notice the relevant API commands to return to the current system state are listed under Commands: You can manually edit this list to add/remove/change commands as desired. Hit the Exit key when done creating the preset.



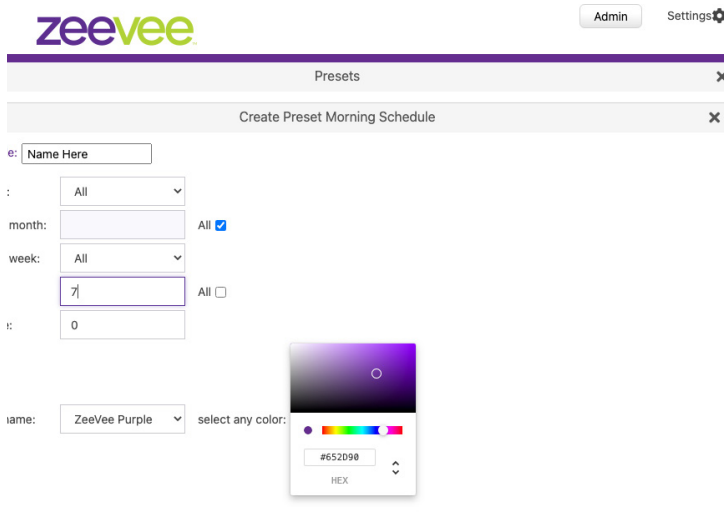
7. The created preset is now available. Options include, running the preset manually, editing the preset, scheduling the preset, showing a log of when preset was run and deleting the preset.



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.



- 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.

- This preset will now appear in the Calendar page.

- 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.

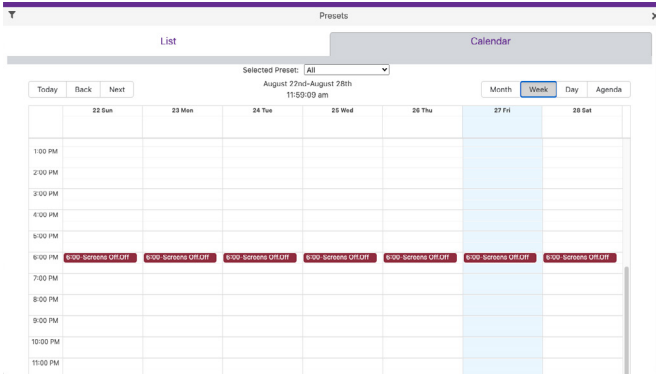
**Note:** The following API commands are used to manually configure NTP servers or set the time for customers without Internet access or a local time server:

```
set server date ntpServer address <domainName>
```

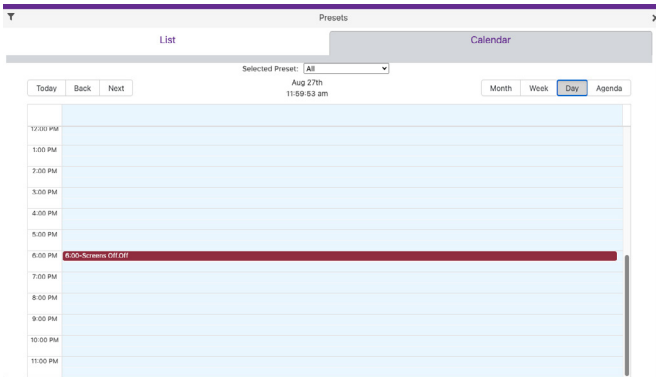
```
set server date manual month <int> day <int> year <int> hour <int> minute <int>
```

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

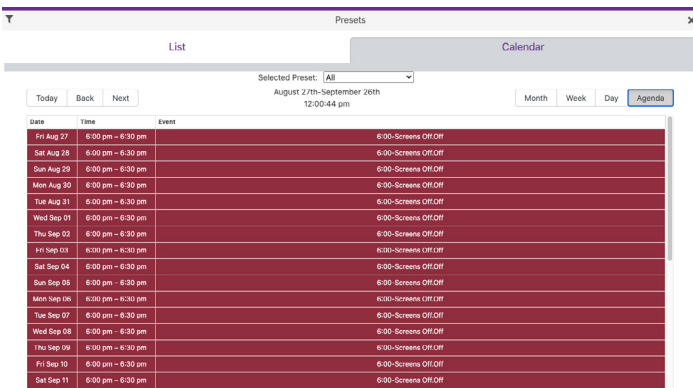
Week Mode shown below:



Day Mode shown below:



Agenda Mode shown below:

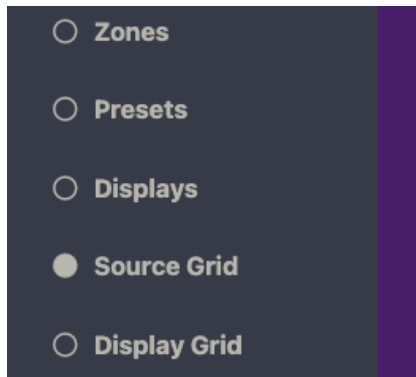


## Source Grid Page

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 ZMP \(page 11\)](#) 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, VideoIn, Network, EDID, Firmware and RS-232. You may want to close other windows to be able to maximize the size of the grid.

The screenshot shows the ZeeVee ZyPer Management Platform interface. The top navigation bar includes the ZeeVee logo, a status indicator (2), the platform name, and 'Admin' and 'Settings' buttons. The main content area is titled 'Source Grid(6)' and features a tabbed interface with 'Status' selected. Below the tabs is a table of encoder information.

Icon	Name	Model	Uptime D:H:M:S	Diag	Resolution/ Status	Gbps
	UHDenc1	ZyperUHD	0-0-0:20	Diag	no link	0.0
	DanteEnc1	ZyperUHD	0-0-0:20	Diag	1920x1080p60.0i	0.4
	Budapest	Zyper4K	0-0-0:0	Diag	down	0.0
	USBenc	Zyper4K	0-0-0:0	Diag	down	0.0
	ZVconReceiver	Zyper4K	0-0-0:0	Diag	down	0.0
	ZeeVee	Zyper4K	0-0-0:0	Diag	down	0.0

- 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.

Icon	Name	Model	Uptime	Diag	Resolution/Status	Gbps
	UHDenc1	ZyperUHD	0:0:0:20	Diag	no link	0.0
	DanteEnc1	ZyperUHD	0:0:0:20	Diag	1920x1080p60.0i	0.4
	Budapest	Zyper4K	0:0:0:0	Diag	down	0.0
	USBenc	Zyper4K	0:0:0:0	Diag	down	0.0
	ZVconReceiver	Zyper4K	0:0:0:0	Diag	down	0.0
	ZeeVee	Zyper4K	0:0:0:0	Diag	down	0.0

- 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.

Icon	Name	RS-232		IR	
		Pair	Tunnel Port	Pair	Tunnel Port
	UHDenc4	Set ✓	none	Set ✓	none
	DanteEnc1	none	none	none	none
	Budapest	none	none	none	none
	USBenc	none	none	none	none
	ZVconReceiver	none	none	none	none
	ZeeVee	none	none	none	none

- The Config tab will let you change encoder names and manually restart or factory default the encoders. 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. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

Icon	Name	Model	Actions		Port Mode	
			Restart	Defaults	Video	Analog Audio
	UHDenc4	ZyperUHD	Restart	Defaults	Set ✓	Set ✓
	DanteEnc1	ZyperUHD	Restart	Defaults	Set ✓	Set ✓
	Budapest	Zyper4K	Restart	Defaults	Set ✓	Set ✓
	USBenc	Zyper4K	Restart	Defaults	Set ✓	Set ✓
	ZVconReceiver	Zyper4K	Restart	Defaults	Set ✓	Set ✓
	ZeeVee	Zyper4K	Restart	Defaults	Set ✓	Set ✓

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.

Source Grid(6) ✕

☰

Status   Routing   Config   **VideoIn**   Network   EDID   Firmware   RS-232

Name	STD	Resolution			Detailed Timing									
		ActiveX	ActiveY	FPS	TotalX	TotalY	PorchX	PorchY	SynX	SynY	PolX	PolY	Quant	
UHDenc4	no link	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
DanteEnc1	VIC16	1920	1080	60.00	2200	1125	88	4	44	5	Pos	Pos	default	
Budapest	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
USBenc	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
ZVconReceiver	VIC16	1920	1080	60.00	2200	1125	88	4	44	5	Pos	Pos	limited	
ZeeVee	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	

Reset   Apply

8. The Network tab has details such as IP address and Subnet Mask. The user can also enable or disable the 1Gb Utility Port on the ZyPer4K. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

Source Grid(6) ✕

☰

Status   Routing   Config   VideoIn   **Network**   EDID   Firmware   RS-232

Name	MAC Address	IP Mode	IP Address	IP Mask	IP Gateway	Utility Port
UHDenc4	34:1b:22:80:7f:3c	linkLocal	169.254.5.13	255.255.0.0	169.254.0.254	NA
DanteEnc1	34:1b:22:81:2b:5c	linkLocal	169.254.4.46	255.255.0.0	169.254.0.254	NA
Budapest	80:1f:12:4d:61:df	dhcp	169.254.224.97	255.255.0.0	NA	enabled
USBenc	80:1f:12:4d:9b:6c	dhcp	169.254.108.155	255.255.0.0	NA	disabled
ZVconReceiver	80:1f:12:4e:56:ab	dhcp	169.254.172.86	255.255.0.0	NA	disabled
ZeeVee	80:1f:12:4e:59:0	dhcp	169.254.1.89	255.255.0.0	NA	disabled

Reset   Apply

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.

Source Grid(6) ✕

☰

Status   Routing   Config   VideoIn   Network   **EDID**   Firmware   RS-232

Name	Filename	Audio	Name	Preferred Res	Max Res	Max Clock	Audio
UHDenc4	Set ✓	Set ✓	zUhd30	3840x2160p30.0C	3840x2160p30.0	300.00	PCM
DanteEnc1	auto	default	zUhd30	3840x2160p30.0C	3840x2160p30.0	300.00	PCM
Budapest	zyper4k60	default	z4k60	4096x2160p60.0C	4096x2160p60.0	600.00	PCM
USBenc	auto	default	z4k60HdR	4096x2160p60.0C	4096x2160p60.0	600.00	PCM
ZVconReceiver	zyper4k60	default	z4k60	4096x2160p60.0C	4096x2160p60.0	600.00	PCM
ZeeVee	zyper4k60-hdr	default	z4k60HdR	4096x2160p60.0C	4096x2160p60.0	600.00	PCM

Reset   Apply

Source Grid(6)

Status Routing Config Videoln Network **EDID** Firmware RS-232

Name	Filename	Audio	Name	Preferred Res	Max Res	Max Clock	Audio
UHDenc4	Set	Set		zyper4k50-hdr-hbraudio			
DanteEnc1	auto			zyper4k60		0.00	PCM
Budapest	zyper4k60			zyper4k60-420		0.00	PCM
USBenc	auto			zyper4k60-420_hdmi14		0.00	PCM
ZVconReceiver	zyper4k60			zyper4k60-hbraudio		0.00	PCM
ZeeVee	zyper4k60-hdr			zyper4k60-hd-hdr		0.00	PCM
				zyper4k60-hdr		0.00	PCM
				zyper4k60-hdr-bt2020		0.00	PCM

Reset Apply

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.

Source Grid(6)

Status Routing Config Videoln Network EDID **Firmware** RS-232

Name	Model	Revision	Update	Update Status	File	% Complete
UHDenc4	ZyperUHD	1.1.2.0	Update	idle	none	0
DanteEnc1	ZyperUHD	1.2.5.0	Update	idle	none	0
Budapest	Zyper4K	4.0.1.0	Update	idle	none	0
USBenc	Zyper4K	4.0.1.0	Update	idle	none	0
ZVconReceiver	Zyper4K	4.0.1.0	Update	idle	none	0
ZeeVee	Zyper4K	4.0.1.0	Update	idle	none	0

Reset Apply

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.

Source Grid(6)

Status Routing Config Videoln Network EDID Firmware **RS-232**

Name	Baud Rate	Data Bits	Stop Bits	Parity	Term Chars
UHDenc4	115200	8	1	none	\x0A\x0D
DanteEnc1	38400	8	1	none	\x0A\x0D
Budapest	2400	8	1	none	\x0A\x0D
USBenc	57600	8	1	none	\x0A\x0D
ZVconReceiver	57600	8	1	none	\x0A\x0D
ZeeVee	115200	8	1	none	\x0A\x0D

Reset Apply

## 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.

The screenshot shows the 'Source Grid(6)' interface with the 'Videon' tab selected. A menu is open over the 'Fi' tab, showing options: 'Show/Hide Columns', 'Export/Import', and 'Help'. The main table displays columns for Name, STD, Resolution (ActiveX, ActiveY, FPS), Detailed Timing (TotalX, TotalY, PerchX, PerchY, SyncX, SyncY, PolX, PolY, Quant), and various status indicators. Below the table are 'Reset' and 'Apply' buttons.

Select the Export/Import option.

The dialog box has a 'Load CSV data file:' field with a 'Drop CSV text file or Click here' placeholder. Below it is a checked checkbox 'Toggle All Cols Selected'. The main table is divided into 'Network' and 'Config' sections. The 'Network' section has columns for MAC Address, IP Mode, IP Address, IP Mask, IP Gateway, Utility Port, Icon, and Name. The 'Config' section has columns for Port Mode, Video, and Analog. Below the table are 'Cancel', 'Export CSV', and 'Import To Grid' buttons.

Network							Config		
MAC Address	IP Mode	IP Address	IP Mask	IP Gateway	Utility Port	Icon	Name	Video	Analog
34:1b:22:80:7f:3d	linkLocal	169.254.5.13	255.255.0.0	169.254.0.254		SatelliteReceiv	UHDenc4	hdmi	
34:1b:22:81:2b:53	linkLocal	169.254.4.46	255.255.0.0	169.254.0.254		SatelliteReceiv	DanteEnc1	hdmi	
80:1f:12:4d:61:df	dhcp	169.254.224.97	255.255.0.0		enabled	cbs.png	Budapest	hdmi	
80:1f:12:4d:9b:6b	dhcp	169.254.108.155	255.255.0.0		disabled	nf1Network.jpg	USBenc	hdmi	
80:1f:12:4e:56:ab	dhcp	169.254.172.86	255.255.0.0		disabled	nbc.png	ZVconReceiver	hdmi	
80:1f:12:4e:59:0	dhcp	169.254.1.89	255.255.0.0		disabled	SatelliteReceiv	ZeeVee	hdmi	

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.

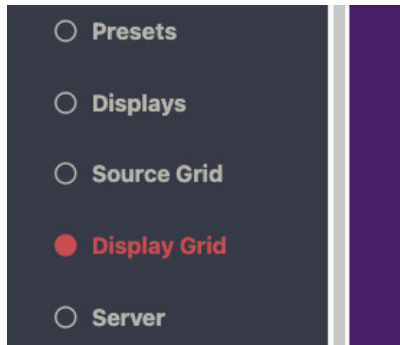


## Display Grid Page

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 ZMP \(page 11\)](#) 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, VideoOut, 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.

Display Grid(5)																		
☰																		
Status				Routing			Config		VideoOut		Network		EDID		Firmware		RS-232	
Icon	Name	Uptime D:H:M:S	Diag	HDMI Video					HDMI Port Audio									
				Source	Mode	Resolution	Rcv	Gbps	Source	Type	Rcv							
🟢	Decoder_1	0:1:56:44	Diag	DanteEnc1	FS	3840x2160p30.0	yes	0.4	video	hdmi	yes							
🔴	UHDdec1	0:0:0:0	Diag	UHDenc4	FS	NA	no	0.0	video	hdmi	no							
🟡	USBdec	0:0:14:8	Diag	Budapest	wallFa	NA	no	0.0	Budapest	native	no							
🔴	ZVconnTrans	0:0:0:0	Diag	Budapest	wallFa	NA	no	0.0	video	hdmi	no							
🔴	BenqDisplay	0:0:0:0	Diag	Budapest	wallFa	NA	no	0.0	video	hdmi	no							

- 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.

Display Grid(5)

☰

Status Routing Config VideoOut Network EDID Firmware RS-232

Icon	Name	Uptime D:HEM:S	Diag	HDMI Video				HDMI Port Audio			
				Source	Mode	Resolution	Rcv	Obps	Source	Type	Rcv
🟡	Decoder_1	0:0:4:50	Diag	NA	none	NA	no	0.0	DanteEnc1	analog	no
🟢	UHDdec1	0:0:0:0	Diag	NA	none	NA	no	0.0	none	hdmi	no
🟢	USBdec	0:0:6:0	Diag	ZVconnReceiver	FS	3840x2160p30.0	yes	8.8	video	hdmi	yes
🟢	ZVconnTrans	0:0:0:0	Diag	Budapest	wallFa	NA	no	0.0	video	hdmi	no
🔴	BengDisplay	0:0:0:0	Diag	Budapest	wallFa	NA	no	0.0	video	hdmi	no

- 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.

Display Grid(5)

☰

Status Routing Config VideoOut Network EDID Firmware RS-232

Icon	Name	Video			HDMI Port Audio		Analog Port Audio		RS-232		IR		
		FastSwitched	Genlocked	Video Wall	digital	analog	digital	analog	USB	Pair	Tunnel Port	Pair	Tunnel Port
🟡	Decoder_1	Set ✓	Set ✓	Set ✓	none	Dante...	Set ✓	Set ✓	Set ✓	Set ✓	none	none	none
🟢	UHDdec1	none	none	NA	none	none	none	none	none	none	none	none	none
🟢	USBdec	none	none	NA	none	none	none	none	none	none	none	none	none
🟢	ZVconnTrans	none	none	NA	video	none	none	none	NA	none	none	none	none
🔴	BengDisplay	none	none	NA	video	none	none	none	NA	none	none	none	none

- 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. You can also change the ZyPerUHD 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:** IdleImage must be 1280x720 in size and will output from decoder at that resolution.

Display Grid(5)

☰

Status Routing Config VideoOut Network EDID Firmware RS-232

Icon	Name	Model	IdleImage	Actions			Port Mode	
				Restart	Defaults	Video	Analog Audio	
☑	Set ✓		Set ✓	Set ✓	Set ✓	Set ✓	Set ✓	
🟡	Z4K-XSenc	Zyper4KX	NA	Restart	Defaults	hdmi	hdmi out	
🟡	Decoder_1	ZyperUHD	IdleImage	Restart	Defaults	hdmi	analogAudio	
🟡	UHDdec1	ZyperUHD	IdleImage	Restart	Defaults	hdmi	hdmi out	
🟡	Z4Kdec1	Zyper4K	NA	Restart	Defaults	hdmi	hdmi out	
🟡	ZVconnTrans	Zyper4K	NA	Restart	Defaults	hdmi	hdmi out	

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.

Display Grid(5)

Status Routing Config **VideoOut** Network EDID Firmware RS-232

Name	Resolution Source	Mode	STD	Resolution			Detailed Timing										
				ActiveX	ActiveY	FPS	TotalX	TotalY	PorchX	PorchY	SyncX	SyncY	PolX	PolY	Quant		
Decoder_1	edid	stretch	no link	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
UHDdec1	edid	stretch	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
USBdec	edid	stret...	VIC95	3840	2160	30.00	4400	2250	176	8	88	10	Pos	Pos	limited		
ZVconnTrans	source	stret...	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
BenqDisplay	edid	box	down	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Reset Apply

8. The Network tab has details such as IP address and Subnet Mask. The user can also enable or disable the 1Gb Utility Port on the ZyPer4K. Note that clicking the 3 horizontal lines in the top right will allow you to show or hide columns.

Source Grid(6)

Status Routing Config VideoIn **Network** EDID Firmware RS-232

Name	MAC Address	IP Mode	IP Address	IP Mask	IP Gateway	Utility Port
UHDenc4	34:1b:22:80:7f:3c	linkLocal	169.254.5.13	255.255.0.0	169.254.0.254	NA
DanteEnc1	34:1b:22:81:2b:8:	linkLocal	169.254.4.46	255.255.0.0	169.254.0.254	NA
Budapest	80:1f:12:4d:61:dff	dhcp	169.254.224.97	255.255.0.0	NA	enabled
USBenc	80:1f:12:4d:9b:6f	dhcp	169.254.108.155	255.255.0.0	NA	disabled
ZVconnReceiver	80:1f:12:4e:56:ab	dhcp	169.254.172.86	255.255.0.0	NA	disabled
ZeeVee	80:1f:12:4e:59:0	dhcp	169.254.1.89	255.255.0.0	NA	disabled

Reset Apply

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.

Display Grid(5)

Status Routing Config VideoOut Network **EDID** Firmware RS-232

Name	Save	Name	Preferred Res	Max Res	Max Clock	Audio
Decoder_1	Save	LG Ultra Hi	3840x2160p30.0	3840x2160p30.0	300.00	PCM
UHDdec1	Save	BenQ LCD	3840x2160p60.0	3840x2160p60.0	600.00	PCM
USBdec	Save	LG Ultra Hi	3840x2160p30.0	3840x2160p30.0	300.00	PCM
ZVconnTrans	Save	Wireless_H	1920x1080p60.0	1920x1080p60.0	150.00	PCM
BenqDisplay	Save	BenQ LCD	3840x2160p60.0	3840x2160p60.0	600.00	PCM

Reset Apply

Display Grid(5)

Status Routing Config VideoOut Network **EDID** Firmware RS-232

Name	Save	Name	Preferred Res	Max Res	Max Clock	Audio
Decoder_1	Save	edid_"devname_"_monitorName".bin			00	PCM
UHDdec1	Save				00	PCM
USBdec	Save				00	PCM
ZVconnTrans	Save				00	PCM
BenqDisplay	Save				00	PCM

Enter filename: edid\_"devname\_"\_monitorName".bin  
 Use \*devname\*, \*devmac\* and \*monitorName\* respectively to substitute actual values. All EDIDs will be downloaded as one .zip file.

Cancel OK

Reset Apply

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 horizontal lines in the top right will allow you to show or hide columns.

Display Grid(5)

Status Routing Config VideoOut Network EDID **Firmware** RS-232

Name	Model	Revision	Update	Update Status	File	% Complete
Decoder_1	ZyperUHD	1.1.2.0	Update	idle	none	0
UHDdec1	ZyperUHD	0.10.2.0	Update	idle	none	0
USBdec	Zyper4K	4.0.1.0	Update	idle	none	0
ZVconnTrans	Zyper4K	4.0.1.0	Update	idle	none	0
BenqDisplay	Zyper4K	4.0.1.0	Update	idle	none	0

Reset Apply

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.

Display Grid(5)

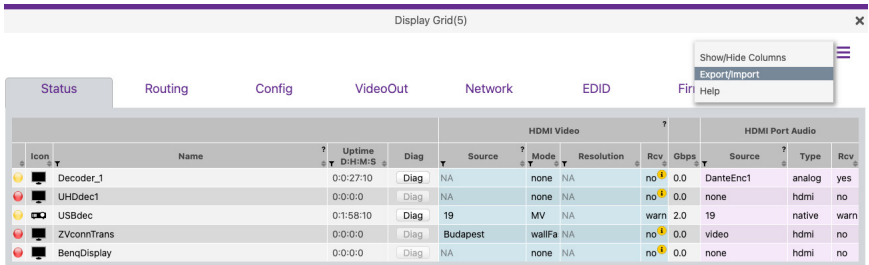
Status Routing Config VideoOut Network EDID Firmware **RS-232**

Name	Baud Rate	Data Bits	Stop Bits	Parity	Term Chars
Decoder_1	57600	8	1	none	\x0A\x0D
UHDdec1	115200	8	1	none	\x0A\x0D
USBdec	57600	8	1	none	\x0A\x0D
ZVconnTrans	57600	8	1	none	\x0A\x0D
BenqDisplay	57600	8	1	none	\x0A\x0D

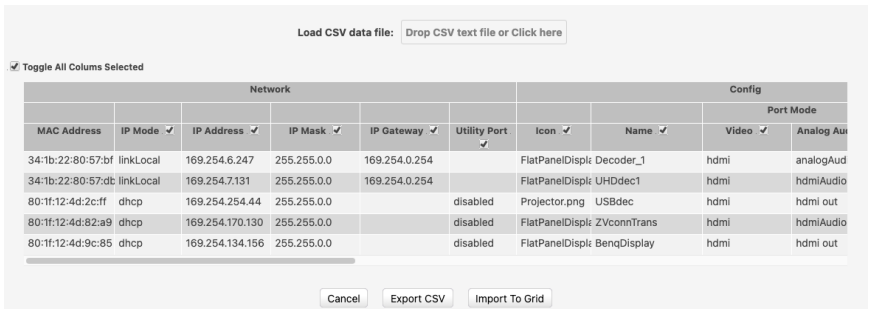
Reset Apply

## 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.

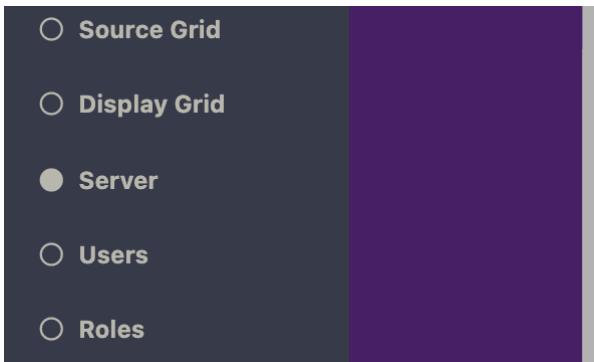
## Server Functions Page

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 ZMP \(page 11\)](#) 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.

Server

### Status

Host Name:	zyper.local
Video Network IP:	172.16.6.111
MAC:	94:c6:91:a2:7a:fc
Version:	2.2.36545
Serial Number:	ZZM1K400007D
Uptime:	0d:1h:32m:39s
Free Memory:	6.79GB
Auto EDID Mode:	Enabled

---

## 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 ZMP \(page 11\)](#) for more information.
2. Click the **Server** tab at the left of the page.
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.

**Note:** There is a check box at the bottom of the Server window for “[Show advanced controls](#)” This box must be checked to enabled changing of the EDID mode setting.

The screenshot displays the 'Server' configuration page. At the top, there is a purple header bar with the word 'Server' in white. Below this, the 'Status' section is highlighted in purple. It contains a table of system information:

Host Name:	zyper.local
Video Network IP:	172.16.6.111
MAC:	94:c6:91:a2:7a:fc
Version:	2.2.36545
Serial Number:	ZZM1K400007D
Uptime:	0d:1h:36m:3s
Free Memory:	6.79GB
Auto EDID Mode:	Enabled

Below the status section, the 'Config' section is highlighted in purple. It contains a sub-section titled 'Encoder Default For EDID Audio' with two radio button options:

- Only PCM
- Allow Compressed

## 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 ZMP \(page 11\)](#) for more information.
2. Click the **Server** tab at the left of the page.
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.

### Status

Host Name:	zyper.local
Video Network IP:	172.16.6.111
MAC:	94:c6:91:a2:7a:fc
Version:	2.2.36545
Serial Number:	ZZM1K400007D
Uptime:	0d:1h:42m:39s
Free Memory:	6.79GB
Auto EDID Mode:	Disabled

### Config

**Encoder Default For EDID Audio**

Only PCM  
 Allow Compressed

### Actions

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 ZMP \(page 11\)](#) for more information.
2. Click the **Server** tab at the left of the page.
3. Scroll down to the License option..

Server

**License**

Product ID:	354CAA85-E826-41BC-6878-94
License:	<input type="text" value="none"/>
Limit:	24
Devices Exceeded:	0

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

---

## Help Page

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

1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Help** tab at the left of the page.
3. The Management Platform **Help** window will be displayed.

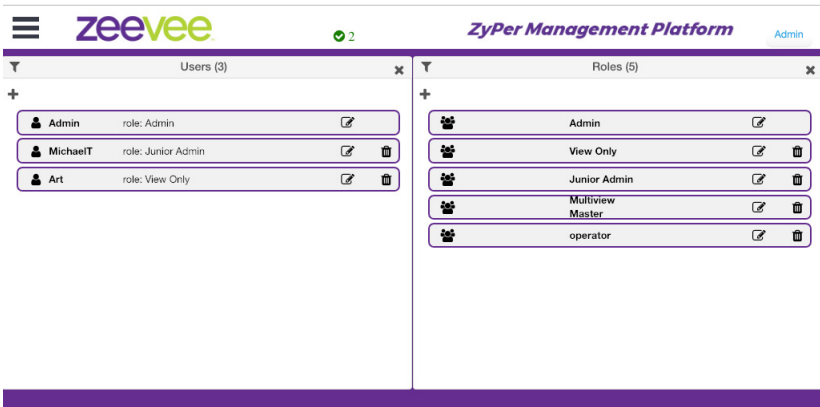
<b>Contents</b>	
What's New in the GUI - Release 2.2	
1 Sources	
2 Displays	
3 Join Configurations	
3.1 What are Join Configurations	
3.2 Pre-defined Join Configurations	
3.3 How are they used	
3.4 Creating a new Join Configuration	
3.5 Modify a Join Configuration	
3.6 Delete Join Configuration	
3.7 Fast Switched vs. Genlocked	
3.8 Resetting default configurations	
4 Preview Streams	
5 Video Walls	
6 Multiview	
7 Zones	
8 Presets	
9 Source Grid	
10 Display Grid	
11 Server	
12 Users	
13 Roles	
14 Logs	
15 How to...	

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

## Users and Roles Pages

The **Users** 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.

1. Login to the ZMP. Refer to [Accessing ZMP \(page 11\)](#) for more information.
2. Click the **Users** tab and **Roles** tab at the left of the page.
3. The Management Platform **Users** and **Roles** windows will be displayed.



The first User shown in the Users 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 edit icon within the Admin user. The following screen will appear and allow you to change the password.



The screenshot shows the Zeevee user management interface. At the top, there is a purple header with the Zeevee logo and a notification icon showing '2'. Below the header is a 'Users' window with a close button. Inside this window is an 'Editor - Admin' form, also with a close button. The form contains the following fields:

- Full Name:** A text input field containing the text 'Admin'.
- User ID:** A text input field containing the text 'admin'.
- Password:** A password input field with a red background, indicating it is required or has an error. The text '(At least one character of any type)' is displayed below the label.
- Password Verification:** A password input field with a red background, also indicating it is required or has an error.
- Force password change on next sign in
- Save:** A button located in the bottom right corner of the form.

Enter the new password on both the "Password" and "Password Validation" lines and then hit the "Save" button in the lower right corner.

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

---

## Adding a New User

To create a new user click on the “+” symbol in the **Users** window. The screen shown below will appear. Enter the Full Name, User ID and Password for the new user. You can keep this password or force the user to generate a new password the first time they login by clicking the “Force password change on next sign in” box.

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

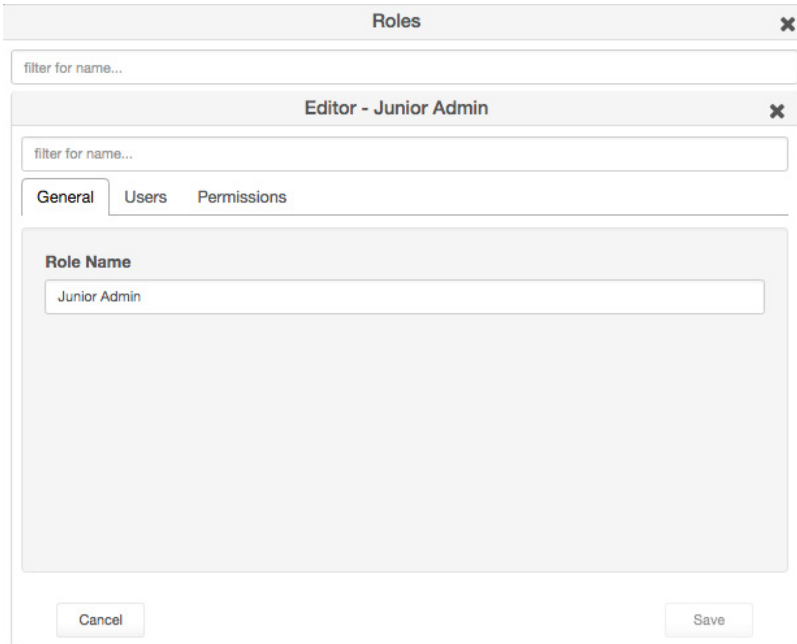
The screenshot shows a web application window titled "Users" with a close button (X) in the top right corner. Inside this window is a sub-window titled "Editor - Sample User" also with a close button (X). The sub-window contains the following elements:

- Full Name:** A text input field containing "Sample User".
- User ID:** A text input field containing "samuser".
- Password:** A password input field with a green background and masked characters "\*\*\*\*\*". A note below the field reads "(At least one character of any type)".
- Password Verification:** A second password input field with a green background and masked characters "\*\*\*\*\*".
- Force password change on next sign in:** A checkbox that is currently unchecked.
- Buttons:** "Cancel" and "Save" buttons are located at the bottom of the sub-window.

---

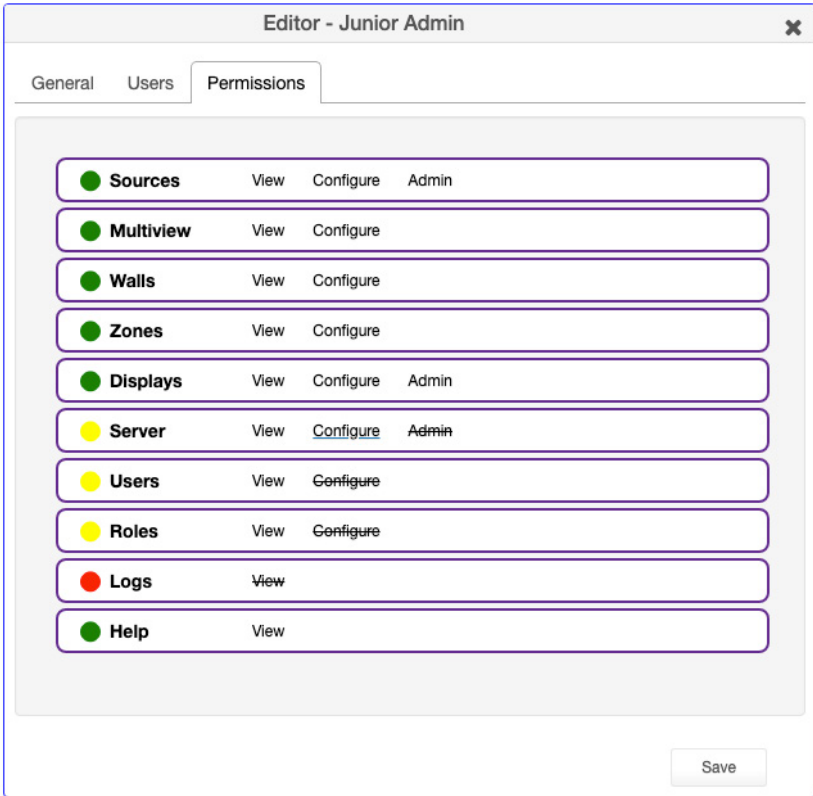
## Adding or Editing a Role

To create a new role click on the “+” symbol in 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”.



The screenshot displays a web interface for managing roles. At the top is a window titled "Roles" with a close button (X). Below it is a search bar labeled "filter for name...". The main content area is a form titled "Editor - Junior Admin" with its own close button. Inside this form, there is another search bar "filter for name...". Below the search bar are three tabs: "General", "Users", and "Permissions". The "General" tab is selected. Under the "General" tab, there is a section titled "Role Name" with a text input field containing the text "Junior Admin". At the bottom of the form are two buttons: "Cancel" and "Save".

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



There are 10 categories that can be assigned different levels of access within the Permissions tab. Sources, Displays, Zones, Walls, Multiview, Logs, Help, Users, Roles and Server. Most categories have 2 or 3 levels of access:

**View:** User is allowed to access the designated tab within ZMP

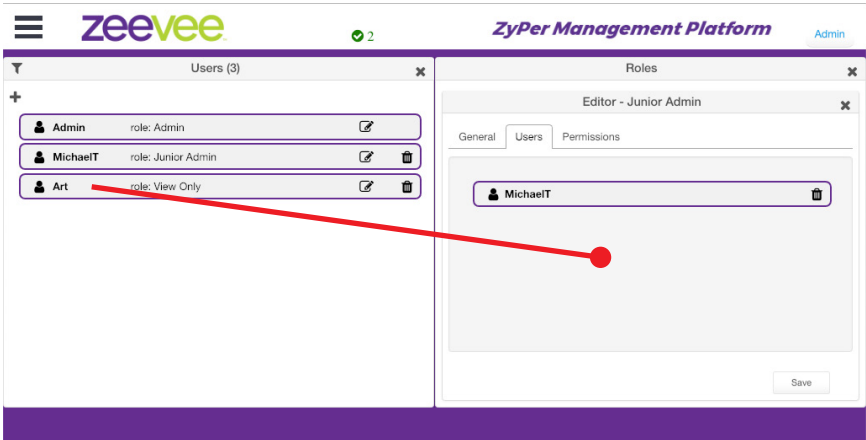
**Config:** User is allowed to alter or create new configurations within the designated tab. For example the user can create a new Wall, Multiview or Zone.

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

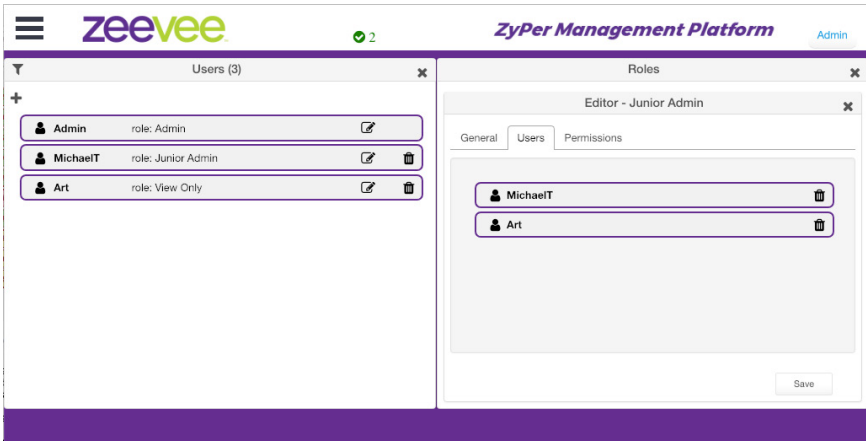
Click on the appropriate levels of access to set permission. The process is color coded to make it easier. Green = Full access, Yellow = Partial access, Red = No access.

Be sure to Save any changes before moving to the step of assigning specific users these permission levels.

You need to have both the **Users** and **Roles** windows visible to assign a User with a set of Roles / Permissions. Open the Users tab in the Roles window and drag a specific User from the User window over. (Drag the “person” icon located to the far left next to user ID)



Be sure to Save any updates.



In example above, “Art” was dragged over to “Junior Admin” role. Be sure to Save this setting before closing the **Roles** window.



# 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 192.168.1.6
```

*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 groups
- `help <group> --` lists commands within a group.

**Note:** The same command may appear in more than one group.

- `help all byGroup` – lists all groups and all commands in each group
- `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 Groups
  Audio
  CEC
  Data
  Decoder/Display
  Device
  Diagnostics
  EDID
  Encoder/Source
  Events
  HDCP
  Join
  Multicast
  Multiview
  Preset
  PreviewStreams
  Redundancy
  SNMP
  Script
  Serial/IR
  Server
  Status/Config
  USB
  Video
  VideoWall
  Zone
Enter 'help <group>', or 'help all byGroup', or 'help all
alphabetical'
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 <encoderMac|encoderName>|none <decoderMac|decoderName|zoneName|.zoneName] > analogAudio
join <encoderMac|encoderName>|videoSource|none <decoderMac|decoderName|.zoneName|.zoneName] > hdmiAudio
join <encoderMac|encoderName|multiviewName>|none <decoderMac|decoderName> multiview
join <encoderMac|encoderName>|none <decoderMac|decoderName|zoneName|.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>
Zyper$
```

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

**(Note: Feature added to release 2.3.37234 and newer)**

```
Zyper$ help search 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 audio
join <encoderMac|encoderName>|none <decoderMac|decoderName|zoneName|.zoneName]> analogAudio
join <encoderMac|encoderName>|videoSource|none <decoderMac|decoderName|.zoneName|.zoneName]> hdmiAudio
set encoder <encoderMac|encoderName> analogAudioOut source none|hdmiAudioDownmix
set encoder <encoderMac|encoderName> edid audio onlyPcm|allowCompressed|serverDefault
set decoder <decoderMac|decoderName> analogAudioOut source analogAudio|hdmiAudioDownmix
set decoder <decoderMac|decoderName> hdmiAudioOut source analogAudio|hdmiPassthroughAudio|hdmiAudio|hdmiAudioDownmix
set multiview <multiviewName> audioSource windowNumber <int>|none
set server encoderDefault edid audio onlyPcm|allowCompressed
start encoder <encoderMac|encoderName> stream video|videoScaled|hdmiAudio|analogAudio
stop encoder <encoderMac|encoderName> stream video|videoScaled|hdmiAudio|analogAudio
Success
```

```
Zyper$ help search create
create multiview <newMultiviewName>
create presetNew <newPresetName> commands existingConnections|empty
create presetSchedule <presetName> schedule <newPresetScheduleName>
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 192.168.1.6
```

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](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 date manual` command.

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

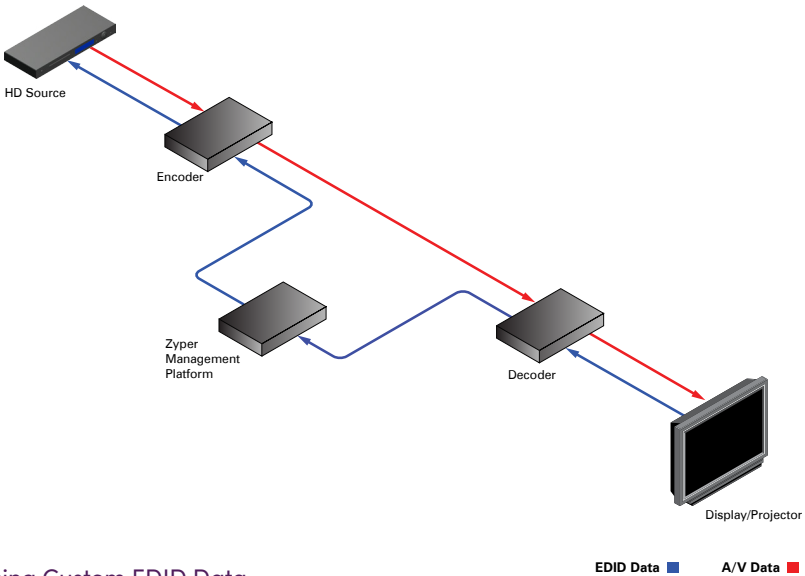
Use the `show server info` command to verify the correct time zone has been set.

```
Zyper$ show server info
server(192.168.1.78);
  server.gen; hostname=zyper.local, version=2.0.35220, previousVersion=2.0.35125, macAddress=1c:1b:0d:82:ff:1a,
  serialNumber=ZZM1H500032B
  server.gen; uptime=0d:0h:13m:13s, freeMem=7.007GB, bootCount=6
  server.gen; runningInVm=false
  server.time; time=Thu Apr 25 22:15:34 2019, timezone=America/
  New_York
  server.license; productID=031B021C-040D-0582-FF06-1A0700080009,
  license=JSGH-RLUH-0000-116F-9328-F426-4BB5-89E2-024D-8CBE-FF1C
  server.license; limit=unlimited, knownDevices=16, devicesUp=0,
  devicesExceeded=0
  server.deviceUpdates; active=0
Success
Zyper$
```

## 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 \(page 84\)](#) 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 192.168.1.6
```

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

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

3. Use the `save device-edid` 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 supplied 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: "vall", 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.

## 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
Encoded Audio	AC3 or other encoded audio formats are passed from encoder to decoder (New in 2.0 release)	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 <b>not</b> supported in Genlocked mode
Video Disconnect	Disconnecting video ( <b>Join None</b> ) will maintain a black screen output. (Video is not technically disconnected)	Disconnecting video ( <b>Join None</b> ) 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
<code>add device</code>	Manually adds a device to Management Platform
<code>add snmp</code>	Add new snmp user or trap server
<code>add zoneDisplay</code>	Adds a display or Video wall to an existing zone.
<code>channel</code>	Cycles up or down through encoders. Used to change channels.
<code>clone multiview</code>	Used to clone an existing multiview
<code>create zone</code>	Creates a new empty zone.
<code>create multiview</code>	Creates a new multiview display (ZyPer4K family only)
<code>create videoWall</code>	Creates an empty 2x2 video wall.
<code>create presetNew</code>	Creates a new preset
<code>create presetSchedule</code>	Creates schedule for existing preset
<code>dataConnect</code>	Creates a TCP port connection between devices for IR or RS232
<code>delete allConfiguration</code>	Deletes all encoder/decoder and server information from the Management Platform
<code>delete device</code>	Deletes the specified encoder or decoder from the Management Platform database.
<code>delete multiview</code>	Deletes the specified multiview from the Management Platform database. (ZyPer4K family only)
<code>delete multiviewWindow</code>	Deletes a specific window from an existing multiview (ZyPer4K family only)
<code>delete preset</code>	Used to delete a preset, preset runlog or preset schedule
<code>delete snmp</code>	Delete SNMP user or trap server
<code>delete videoWall</code>	Deletes the specified video wall from the Management Platform database.
<code>delete zone</code>	Deletes an existing zone
<code>delete zoneDisplay</code>	Removes a display from an existing zone
<code>diagnostics device</code>	Runs a set of diagnostics on device
<code>dumpusb</code>	Outputs information about USB devices
<code>events</code>	Causes the event mode to be entered
<code>factoryDefaults device</code>	Sets the specified encoder/decoder to factory-default settings.
<code>flashLeds</code>	Physically identifies the specified encoder/decoder on the network using LED flashes. (ZyPer4K family only)
<code>join</code>	Switches audio and/or video from source to display or video wall
<code>join videoSource</code>	Selects audio feed to follow a video join

Command	Description
<code>load encoderEdid</code>	Uploads an EDID file to the specified encoder
<code>load idleImage</code>	Uploads an image to use at UHD background when no video streamed to decoder
<code>logging</code>	Used to set logging level and add notes to the log
<code>previewStream</code>	Used to turn on/off the preview stream viable in the Management Platform GUI (ZyPer4K family and ZyPerUHD only)
<code>redundancy switchover</code>	Swap Management Platform Master and Slave
<code>redundancy delete downServers</code>	Removes no longer present servers from list of redundant servers
<code>restart device</code>	Restarts the specified encoder/decoder
<code>restart server</code>	Restarts the Management Platform
<code>restore server database</code>	Restores a saved database
<code>revert server</code>	Switch to a previously installed version of the API
<code>save deviceEdid</code>	Saves the EDID from a decoder to a local file
<code>save server database</code>	Saves current server database to file
<code>save system config</code>	Saves current system configuration to a file
<code>send</code>	Sends an IR, CEC or RS232 string to the specified device
<code>script</code>	Executes the specified AJAX/JSON or text script.
<code>send</code>	Sends an IR, RS232 or CEC string to the specified device
<code>set encoder analogAudioOut source</code>	Sets the source of Analog audio output for specified encoder (ZyPer4K family only)
<code>set encoder edid audio</code>	Sets allowable input audio formats
<code>set encoder hdcpMode</code>	Sets the HDCP compatibility at the encoder side
<code>set decoder connectionMode</code>	Changes current connection to decoder to fast-switched, genlocked or genlocked-scaled. (ZyPer4K family only)
<code>set decoder displayMode</code>	Sets default decoder output to crop, stretch or box
<code>set decoder displayResolution</code>	Set decoder output size to auto or manual resolution. (Width, Height, FPS)
<code>set decoder analogAudioOut source</code>	Sets the source of Analog audio output for specified decoder
<code>set decoder edidPreferMode</code>	Sets the preferred resolution from the display EDID
<code>set decoder hdmiAudioOut source</code>	Sets the source of HDMI audio output for specified decoder
<code>set decoder hdmi5vControl</code>	Enables or disables HDMI 5V line (ZyPer4K-XS and ZyPer4K-XR only)

Command	Description
<code>set decoder osdStatusMode</code>	Enables or disables OSD feature (ZyPerUHD only)
<code>set decoder powerSave</code>	Enables or disables power-save feature (ZyPerUHD only)
<code>set device utilityPort</code>	Enables or disables the 1G Ethernet utility port for the specified device (ZyPer4K family only)
<code>set device general name</code>	Sets the name for the specified device.
<code>set device ip dhcp linkLocal</code>	Sets the specified device to DHCP or Link-Local mode (ZyPer4K family only)
<code>set device ip static</code>	Sets the device to static mode (ZyPer4K family and ZyPerUHD only)
<code>set device irProcessing</code>	Configures decoder to process incoming IR commands to issue Channel up/down command (ZyPer4K family only)
<code>set device rs232</code>	Sets the RS232 settings for the specified device
<code>set device sendIpMcastRange</code>	Sets allowable range of multicast addresses for selected devices (ZyPer4K family only)
<code>set device sourceDisplay iconImageName</code>	Sets the icon image for the specified device.
<code>set device sourceDisplay location</code>	Sets the location name for the specified device.
<code>set device sourceDisplay manufacturer</code>	Sets the manufacturer name for the specified device.
<code>set device sourceDisplay model</code>	Sets the model name for the specified device
<code>set device sourceDisplay serialNumber</code>	Sets the serial number name for the specified device
<code>set device usbFilter</code>	Allows restrictions to USB use on selected device (ZyPer4K family only)
<code>set device videoPort</code>	Selects active input port for ZyPer4K units with multiple inputs (ZyPer4K family only)
<code>set multiview</code>	Assigns source to a position and size within a multiview display (ZyPer4K family only)
<code>set multiview audioSource windowNumber</code>	Selects the input source to provide Audio for multiview display (ZyPer4K family only)
<code>set multiview canvasSize</code>	Specifies Multiview canvas for multiview creation. (ZyPer4K family only)
<code>set multiview newEncoderName</code>	Used to specify a new encoder for existing multiview window. Can also set to "none"
<code>set preset commands</code>	Specifies commands to be used for a preset
<code>set preset description</code>	Sets a description for the preset
<code>set preset schedule eventColor</code>	Sets the color to be used for a preset schedule in the GUI calendar

Command	Description
<code>set preset schedule month</code>	Sets the schedule month/day/time to run preset
<code>set responses rs232TermChars</code>	Specifies the RS232 termination string
<code>set server api lineWrap</code>	Sets number of characters before API command line interface starts a new line
<code>set server autoEdidMode</code>	Sets the EDID mode
<code>set server dataTunnelMode</code>	Sets server transfer mode to raw or telnet
<code>set server date</code>	Used to set server date manually or via ntp server
<code>set server discoverMode</code>	Used to set how server discovers ZyPerUHD endpoints. Broadcast or Multicast
<code>set server ip</code>	Sets the IP address of the Management Platform
<code>set server license</code>	Sets server license. (Max endpoints)
<code>set server isaac address</code>	Sets the domain name of the Isaac server
<code>set server isaac subsystemId</code>	Sets the subsystem ID of the Isaac server
<code>set server redundancy</code>	Set a virtual IP address/mask for Master and Slave Management Platforms
<code>set server ssh password</code>	Used to set ssh password
<code>set server telnet mode</code>	Used to enable or disable telnet access
<code>set server telnet password</code>	Used to set telnet password
<code>set server timezone</code>	Sets the time zone
<code>set terminal output</code>	Select either normal or JSON format output from API
<code>set videoWall</code>	Modifies an existing wall
<code>set videoWall Decoder</code>	Assigns the specified decoder to a position within the video wall
<code>show dataTunnels</code>	Shows what rs232 or IR data relay ports are opened on the server.
<code>show device capabilities</code>	Shows detailed capabilities of specified device or devices

Command	Description
<code>show device config</code>	Shows detailed configuration information for specified device or devices
<code>show device connections</code>	Shows encoder connections to decoders
<code>show device status</code>	Provides detailed status information for specified device or devices
<code>show device userAdded</code>	Will show a list of all ZyPer endpoints that have been manually added with the add device command
<code>show multiviews config</code>	Lists all created multiviews with source, position and size info (ZyPer4K family only)
<code>show multiviews status</code>	Lists all created multiviews with source, datarate and multicast address info (ZyPer4K family only)
<code>show preset</code>	Shows information and configuration details for a preset
<code>show previewStreams</code>	Lists names of encoders currently generating a preview stream. (ZyPer4K family and ZyPerUHD only)
<code>show responses</code>	Displays the lastChangeld for the specified device
<code>show server config</code>	Displays the IP address and EDID mode of the Management Platform
<code>show server info</code>	Displays Management Platform information
<code>show server redundancy</code>	Displays information about Master and Slave Management Platforms
<code>show snmp</code>	Displays information related to snmp
<code>show values</code>	Shows information related to encoder, decoder, server and multiviews
<code>show videoWalls</code>	Displays a list of all created video walls
<code>show zones</code>	Displays a list of zone and displays contained within
<code>shutdown server</code>	Shuts down or reboots the Management Platform
<code>sleep</code>	Sets a time delay, in milliseconds
<code>stop encoder</code>	Stop a specified stream (ZyPer4K family only)
<code>start encoder</code>	Start a specified stream (ZyPer4K family only)
<code>switch</code>	Switches IR or RS-232 between devices
<code>troubleReport</code>	Generates a trouble report
<code>update device</code>	Updates the individual encoder or decoder units
<code>update server</code>	Updates the Management Platform software. See <a href="#">Updating the Software (page 259)</a> 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.

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 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 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
```

## 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 existing 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 \(page 41\)](#) 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
```

---

## 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 \(page 41\)](#) 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 250 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 250 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 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 \(page 36\)](#) 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
```

---

## 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.

*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 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. 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 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 decoder 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 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 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
```

---

## 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 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 fiber LED indicators on the device will flash for 5 seconds. (ZyPer4K family only)

### 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
```

## 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. In order to control what type of audio is being output from the decoder, refer to the `set decoder AnalogAudioOut` command.
- ▶ **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 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
<code>analogAudio</code>	Embed audio from the specified encoder
<code>fastSwitched</code>	Join in "fast-switched" mode
<code>genlocked</code>	Low-latency join mode (ZyPer4K family only)
<code>genlockedScaled</code>	Low-latency with scale up/down (ZyPer4K family only)
<code>hdmiAudio</code>	Join hdmi-audio to either hdmi-out or analog-out. Note this command will cause <code>hdmiAudioFollowVideo=False</code> for specified decoder. See <code>join videoSource</code> command on next page.
<code>multiview</code>	Join a multiview to a display (ZyPer4K family only)
<code>videoWall</code>	Join a source to a video-wall
<code>video</code>	Join video only (audio not joined)
<code>window</code>	Join any portion of a source to any portion of a display (ZyPer4k family only)
<code>usb</code>	Establish USB connection
<code>"none"</code>	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 hdmi-audio  
Success
```

```
join myMultiview2 Display4 multiview  
Success
```

```
join myEncoder1 myWall videoWall  
Success
```

```
join none myDecoder1 fastSwitched
```

## 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 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.

### Build in EDID options

```
zyper-default
zyper4k25
zyper4k30
```



---

## Build in EDID options continued

```
zyper4k50
zyper4k50-420
zyper4k50-420_hdmi14
zyper4k50-hbraudio
zyper4k50-hd-hdr
zyper4k50-hdr
zyper4k50-hdr-bf2020
zyper4k50-hdr-bf2020-hbraudio
zyper4k50-hdr-hbraudio
zyper4k60
zyper4k60-420
zyper4k60-420_hdmi14
zyper4k60-hbraudio
zyper4k60-hd-hdr
zyper4k60-hdr
zyper4k60-hdr-bf2020
zyper4k60-hdr-bf2020-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-bf2020
zyperUhd50-hdr-bf2020-hbraudio
zyperUhd50-hdr-hbraudio
zyperUhd60
zyperUhd60-420
zyperUhd60-420_hdmi14
zyperUhd60-hbraudio
zyperUhd60-hd-hdr
zyperUhd60-hdr
zyperUhd60-hdr-bf2020
zyperUhd60-hdr-bf2020-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

```
set decoder osdStatusMode
```

---

## 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
```

## previewStream

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

---

## 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
```

---

## 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.

### Example

```
restart device myEncoder2  
Success
```

```
restart device 0:1e:c0:f6:cb:76  
Success
```

### Related Commands

```
restart server
```



---

## restart server

Restarts the server. Linux is not restarted.

### Syntax

```
restart server
```

### Parameters

*none*

### Example

```
restart server  
Success
```

### Related Commands

[restart device](#)

---

## 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.
yper
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 in standard format. The `.txt` file is the decoded EDID data. See [Using Custom EDID Data \(page 79\)](#) 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. (Not supported on ZyPerUHD)										
cec	on   off (Used to turn a device on or off)										
cec hexString	hex-numerals-no-delimiters (ZyPer4K family 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: <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;">\n</td> <td>New line</td> </tr> <tr> <td style="padding-right: 20px;">\r</td> <td>Carriage return</td> </tr> <tr> <td style="padding-right: 20px;">\t</td> <td>Tab</td> </tr> <tr> <td style="padding-right: 20px;">\\</td> <td>Slash</td> </tr> <tr> <td style="padding-right: 20px;">\xnn</td> <td>Hex value, where nn is a two-digit hex value, including leading zeros</td> </tr> </table>	\n	New line	\r	Carriage return	\t	Tab	\\	Slash	\xnn	Hex value, where nn is a two-digit hex value, including leading zeros
\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 0000006900000015005f001700300017003000170030001
700300017001700170030001700170017001700170030001700170017003000170
03000170017001700300017001700170017001700170030001700300017003
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 encoder analogAudioOut

Sets the analog audio output source type for the specified encoder. (ZyPer4K family only)

### 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 audio output to use.

argument	Description
analogAudioOut	Audio output from the Audio port on the Encoder.

*type*

Type: **STRING**

The audio mode (analog or HDMI).

argument	Description
source none	No analog audio output from the encoder
source	Uses downmixed audio from input HDMI stream.
hdmiAudioDownmix	

### Example

```
set encoder Myencoder1 analogAudioOut source hdmiAudioDownmix
Success
```

---

## set encoder edid audio

Sets the allowable audio input formats at the encoder. (ZyPer4K and ZyPerUHD only)

### 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
ZyPerHD	Part of fast-switched connection	No Change
ZyPerHD	Part of fast-switched connection	No Change

-----  
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 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 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 only)
hdmiAudioOut	Audio output from the HDMI port on the decoder. (ZyPer4K family only)

*type*

Type: **STRING**

HDCP options. (Note: Valid with ZyPerUHD 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.
source hdmiAudio	Uses the HDMI stream (HDMI audio-out only) Use if video in Fast-Switch mode.
source hdmiPassthroughAudio	Used if video is in Genlock mode.
source hdmiAudioDownmix	Uses the HDMI-downmix stream.

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 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 ultimately 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 “HDMI 2.0” 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 1.4 Devices

No scaler, effectively always in “displayResolution source” mode. Source is always sent to output. If output can’t handle source, there will be no video.

Decoder Preferred Resolution is only status; it is never used to affect the decoder output resolution. Decoder “displayResolution” overrides are ignored.

## 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
  - Means 1080p60 scaling to UHD must be UHDp60, which won’t work if display is only UHDp30 capable.
  - If UHDp60 > decoder Preferred Resolution, then the output is left at 1080p60.
- If source is 1080i
  - Output must be input FPS \* 2
  - 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 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 of 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/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 osdStatusMode

Enables or disables on-screen-display feature of the decoder. (ZyPerUHD 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 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 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.

*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 iconName  
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 gatew
```

### 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.

*gatew*

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 an 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 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 iconName
set device sourceDisplay location
set device sourceDisplay manufacturer
set device sourceDisplay model
set device sourceDisplay serialNumber
```

---

## 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 iconName
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 iconName 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 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	Allows any USB device except HID devices
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 utilityPort

Enables or disables the 1Gb Utility Ethernet port on the specified encoder or decoder.  
(ZyPer4K 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.

### 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 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 (Located to the right)
hdmiOptionalIn	Use the HDMI input (Located to the left). "Primary Input" Dual input HDMI only.
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
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
```

### Related Commands

```
set device general name
set device ip
set device rs232
```

## 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 myEncl
percentPositionX 50 percentPositionY 50 percentSizeX 25
percentSizeY 25 layer 3
```

### Using Pixel Values

```
set multiview myMview1 windowNumber 1 encoderName myEncl
pixelPositionX 1920 pixelPositionY 1080 pixelSizeX 800 pixelSizeY
600 layer 3
```

## Related Commands

```
create multiview
delete videoWall
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 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 window

### 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 specified 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 specified 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)

Create a string of text to be overlaid 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 associated 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 commands

### 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 sname eventColor color
```

### Parameters

*id*

Type: **STRING**

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

*sname*

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. #22fee), 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 specified, 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 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 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 ZyPerUHD endpoints are discovered by the Management Server on the network

### Syntax

```
set server discoverMode 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

### Notes

Allows the server to discover ZyPerUHD 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 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
ZyPerHD	Part of fast-switched connection	No Change
ZyPerHD	Part of fast-switched connection	No Change

## 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 IP Address Mask Gateway DNS-Server reboot
```

### Parameters

*id*

Type: **STRING**

Supply one of the following arguments.

argument	Description
server	Select the "Video" network. (ZyPer Network)
management	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 server dhcp reboot
Success
```

```
set server ip server static 192.168.1.26 255.255.255.0 none none
reboot
Success
```

```
set server ip management static 192.168.4.20 255.255.255.0
192.168.4.1 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 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 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 ssh password

Sets the password for ssh access.

The ssh account is: zyper  
The default password is: zyper

If a password is not provided, then the current password will be deleted. In this case, no password prompt will be displayed.

### Syntax

```
set server ssh password pass
```

### Parameters

*pass*

Type: **STRING**

The desired password. (String with length from 4 to 64 characters)

Valid Password Characters include: a - z A - Z 0 - 9 ~ ! @ # \$ % ^ \* \_ - = + { }

[ ], . / ? :

Invalid Password Characters include: & ( ) | \ ; ' " `

### Examples

```
set server ssh password biGB055
Success
```

```
ssh zyper@<ip_address>
```

### Notes

To reset system to no ssh 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 password
set server telnet mode
set server timezone
```

---

## 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 ssh password
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 ssh password  
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](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":[],"command":"set terminal output json echo yes prompt yes "}
```

```
set terminal output json echo no prompt yes
[ 112]{"status":"Success","text":[],"errors":[],"warnings":[],"command":"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 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, Maximum 15 for ZyPerUHD, Maximum 4 for ZyPerHD)

*cols*

Type: **INTEGER**

The number of columns. (Max 15 for ZyPer4K, Max 15 for ZyPerUHD, 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
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
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 or MAC address of the encoder. String names are case-sensitive. If `none` is passed as the argument, then the display is disconnected from that position in the video wall.

*name*

Type: **STRING**

The updated name of the video wall.

### Example

```
set videoWall myWall12 newName yourWall12  
Success
```

### Related Commands

```
create videoWall  
set videoWall  
show videoWalls  
join videoWall
```

---

## 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 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. Can also enter in one of the arguments below.

argument	Description
all	Displays configuration information for all available devices.
commands	Shows all the command used to configure every device, multiview, video wall in the system
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 Curved
device(d8:80:39:9a:e6:d);
  device.gen; model=Zyper4K, type=encoder, name=Curved, state=Up,
lastChangeId=124
  device.gen; firmware=4.0.1.0
  device.gen; ethernetManagementPortMode=disabled
  device.hdmi; htcpMode=enabled
  device.ports; videoPort=hdmi
  device.ip; mode=dhcp, address=172.16.6.33, mask=255.255.255.0,
gateway=172.16.6.1
  device.rs232; sendingToMacOrIp=Top-Right(172.16.6.42),
terminationChars=\x0A\x0D, baudrate=57600, dataBit=8, stop_Bit=1,
parity=none
  device.ir; sendingToMacOrIp=Bot_Right(172.16.6.71)
  device.source; iconName=foxSports, manufacturer=none,
model=none, location=none, serialNumber=none
  device.audioOutSourceType; analogOutSourceType=none
  device.edid; loadMode=auto, audio=serverDefault
  device.sendIpMcastRange; first=224.1.1.1, last=239.255.255.255
  device.videoStream; ipMcastAddr=224.1.1.14, mode=enabled
  device.videoScaledStream; ipMcastAddr=224.1.1.1, mode=disabled,
sizeX=0, sizeY=0
  device.analogAudioStream; ipMcastAddr=0.0.0.0, mode=disabled
  device.hdmiAudioStream; ipMcastAddr=224.1.1.16, mode=enabled
  device.previewStream; mode=disabled
lastChangeIdMax(125);
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 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 add 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 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 <i>type</i> parameter.
lastChangeId	Displays the <b>lastChangeId</b> 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 server config

Displays configuration information for the Management Platform.

### Syntax

```
show server config
```

### Parameters

*none*

### Example

```
show server config
server(192.168.0.22);
  server.gen; autoEididMode=enabled, redundancy=enabled
  server.ipServerAddress; mode=static, address=192.168.0.22,
mask=255.255.255.0, gateway=none, dns=none
  server.ipManagementAddress; mode=none, address=NA
  server.ntpServer; address=ntp.ubuntu.com
  server.telnetAccess; mode=enabled
  server.encoderDefault.edid; audio=onlyPcm
  server.dataTunnelMode; telnet=telnetHandshakeMode
  server.logging; level=1
  server.isaac; address=none, subsystemId=none
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
```

### Parameters

*none*

### Example

```
show server info
server(192.168.0.22);
  server.gen; hostname=zyper.local, version=2.3.36999,
previousVersion=2.2.36870,
macAddress=94:c6:91:a0:47:fc, serialNumber=ZZM1K400011D
  server.gen; uptime=3d:21h:25m:24s, freeMem= 6.71GB, bootCount=173
  server.gen; runningInVm=false
  server.ipActive; ipServerAddr=192.168.0.22, ipManagementAddr=NA,
gatewayAddr=none, dnsAddr=none
  server.time; time="Tue Aug 31 08:43:59 2021",
timezone=America/New_York
  server.pollStats; count=0, interval: 0-minutes, monListSize=0
  server.license; productID=F9188182-AF72-C6C8-92C6-94C691A047FC,
license=none
  server.license; Zyper4KLimit=24, Zyper4KDevices=6, allDevices=12,
allDevicesUp=6, Zyper4KDevicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_0.0.0.0=1, num_2.0.4.0=2,
num_4.1.2.0=3
Success
```

### Related Commands

```
show server config
revert server
```

---

## show server redundancy

Displays information about master and slave Management Platforms

### Syntax

```
show server redundancy
```

### Parameters

*none*

### 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 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.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|
hdm|displayPort|hdmioptionalIn|vga|component|composite|s-
video|analogNone|hdsdi
  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.hdcpcMode; 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.downlinks; values=[none] | [mac=<decMac1>|link_1, name
=<decName1>|existsButUnknown], [mac=<decMacN>|link_N, name=<decName
N>|existsButUnknown]
  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
```

---

## 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.
```

### Related Commands

```
restart server
```

---

## 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](#)

## 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`

## 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
<code>ir</code>	Specifies a IR connection. (ZyPer4K family and ZyPerUHD only)
<code>rs232</code>	Connection to another device or the server. Set <code>rxid = none</code> 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 file is in .tgz format and is written to the `/srv/ftp/files` folder on the Management Platform.

### Syntax

```
troubleReport
```

### Parameters

*none*

### Example

```
troubleReport
Creating Trouble Report
Saving device status and configuration...
Saving SQL database...
Saving system files...
Saving rcServerMain binary file...
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_v3_5_2_0.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 \(page 259\)](#) 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-software-file.zyper  
Success
```

```
Server rebooting; connection will end
```

### Important Note:

The ZyPer MP update file will be available in three, platform-specific versions. Please use the correct version for the hardware platform being updated.

ZyPerMP NUC computer: `update_nuc_2.2.xxxx.zyper`  
ZyPerMP Proserver: `update_proserver_2.2.xxxx.zyper`  
ZyPerMP VMware: `update_vm_2.2.xxxx.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"
  - Causes the event mode to be entered
  - Server sends initial events (described below) and new events as they occur to this telnet session
  - 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"); // ignored 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"
lastChangeld(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"
lastChangeld(32);
Success
Zyper$
Zyper$ show events since 32
lastChangeld(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 lastChangeld 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-
80016003500160034001600340016000f001500100015000f001500100015000f0015001000
140010001400100015000f001500340016003300160034001600340016003300160033001600-
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](http://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 still 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 sender 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 Technical 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 your 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

## Appendix

There are three ways to receive events:

---

## 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_2.2.36708.zyper
```

9. Press the `[ENTER]` key. Information similar to the following will be displayed.

```
local: update_nuc_1.07.1.33348.zyper remote: update_
nuc_1.7.1.33348.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_2.2.36708.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_1.7.1.33321.zyper  
Success
```

```
Server rebooting; connection will end
```

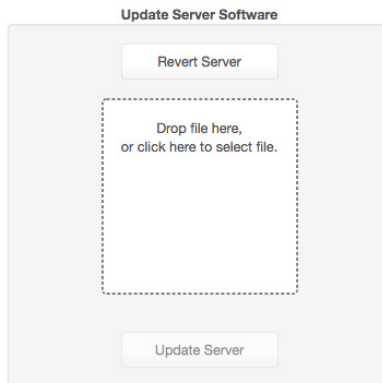
---

## 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 ZMP \(page 11\)](#) 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.



Show advanced controls

---

**Important Note:**

The ZyPer MP update file will be available in four, platform-specific versions. Please use the correct version for the hardware platform being updated.

ZyPerMP NUC computer (Single Ethernet Port): update\_nuc\_2.5.xxxx.zyper  
ZyPerMP NUC computer (Two Ethernet Ports): update\_nuc2004\_2.5.xxxx.zyper  
ZyPerMP Proserver: update\_proserver\_2.5.xxxx.zyper  
ZyPerMP VMware: update\_vm\_2.5.xxxx.zyper

---

## 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.
- 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.
- 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.

Server

**Redundancy**

IP: 172.16.5.240

Virtual IP:

Virtual Mask:

State: Master

**Preferred**

Master

Slave

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)
```

## Hardware Specifications (Intel NUC version)

CPU	<ul style="list-style-type: none"> <li>Intel® Pentium® Processor J5005</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 16.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>Intel® HD Graphics 600</li> </ul>
LAN	<ul style="list-style-type: none"> <li>Gigabit LAN</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>8 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>Input: 100 ~ 240 V AC</li> <li>Output: 19V DC, 3.42 A</li> </ul>
I/O	<ul style="list-style-type: none"> <li>2 x HDMI 2.0a</li> <li>4 x USB 3.0, Type- A, female</li> <li>1 x RJ45</li> <li>1 x 19V DC</li> <li>1 x Kensington lock slot</li> <li>2 x 3.5mm headset jacks (Not used)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>0 °C to +40 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-20 °C to +60 °C</li> </ul>
VESA	<ul style="list-style-type: none"> <li>VESA Bracket included</li> <li>Supports 75 x 75 and 100 x 100 mm</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>4.55 in x 2.01 in x 4.57 in (115 mm x 51 mm x 111 mm)</li> </ul>



RoHS



## Hardware Specifications (NUC version Feb 2022 and beyond)

CPU	<ul style="list-style-type: none"> <li>Intel® Celeron® Processor N3350</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 20.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>Intel® HD Graphics 500</li> </ul>
LAN	<ul style="list-style-type: none"> <li>1x 100mb LAN (Eth0), 1x Gigabit LAN (Eth1)</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>4 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>Input: 100 ~ 240 V AC</li> <li>Output: 19V DC, 3.42 A</li> </ul>
I/O	<ul style="list-style-type: none"> <li>2 x HDMI 2.0a</li> <li>3 x USB 3.0, Type-A, female</li> <li>2 x USB 2.0, Type-A, female</li> <li>2 x RJ45</li> <li>1 x 19V DC</li> <li>1 x Kensington lock slot</li> <li>1 x 3.5mm headset jacks (Not used)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>0 °C to +40 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-20 °C to +70 °C</li> </ul>
VESA	<ul style="list-style-type: none"> <li>VESA Bracket included</li> <li>Supports 75 x 75 and 100 x 100 mm</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>6.06 in x 1.25 in x 4.25 in (154 mm x 32 mm x 108 mm)</li> </ul>



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 (Enterprise Grade Rack Mount)

CPU	<ul style="list-style-type: none"> <li>Intel® Xeon E3-1200 v5</li> </ul>
Operating System	<ul style="list-style-type: none"> <li>Linux Ubuntu 16.04</li> </ul>
Internal Storage	<ul style="list-style-type: none"> <li>64 GB SSD</li> </ul>
Graphics	<ul style="list-style-type: none"> <li>ASPEED AST2400 BMC</li> </ul>
LAN	<ul style="list-style-type: none"> <li>Dual Gigabit LAN</li> </ul>
Internal Memory	<ul style="list-style-type: none"> <li>8 GB DDR4</li> </ul>
Power Supply	<ul style="list-style-type: none"> <li>200W Low-Noise AC-DC power supply. Fan speed dynamically adjusts for load and environment.</li> </ul>
I/O	<ul style="list-style-type: none"> <li>1 x VGA (15-pin D-sub)</li> <li>2 x USB 2.0, Type-A, female</li> <li>2 x RJ45 (LAN) (Video Network and Management Network)</li> <li>1 x RS232 (9-pin D-sub)</li> </ul>
Operating Temperature	<ul style="list-style-type: none"> <li>+10 °C to +35 °C</li> </ul>
Storage Temperature	<ul style="list-style-type: none"> <li>-40 °C to +70 °C</li> </ul>
Dimensions (W x H x D)	<ul style="list-style-type: none"> <li>17.2 in x 1.7 in x 11.3 in</li> <li>(437 mm x 43 mm x 287 mm)</li> </ul>
Ambient Noise	<ul style="list-style-type: none"> <li>Measurement point was 1M distant, straight in front of unit</li> <li>43.5 db(A) - Startup and peak load condition</li> <li>32.8 - 34.4 db(A) - Expected range during typical load</li> <li>31.7 db(A) - Idle and very lightly loaded or cooler ambient conditions</li> </ul>
Weight	<ul style="list-style-type: none"> <li>8.45 lbs, (3.83 kg)</li> </ul>



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	<ul style="list-style-type: none"><li>• Dual CPU's</li></ul>
Server Platform	<ul style="list-style-type: none"><li>• VMware ESXi 6.0 or later</li></ul>
Internal Storage	<ul style="list-style-type: none"><li>• 64 GB or greater</li></ul>
LAN	<ul style="list-style-type: none"><li>• Gigabit LAN</li></ul>
Internal Memory	<ul style="list-style-type: none"><li>• 8 GB or greater</li></ul>





**North America:** +1 (347) 851-7364 • [sales@zeevee.com](mailto:sales@zeevee.com)  
**International:** +1 (347) 851-7364 • [sales@zeevee.com](mailto:sales@zeevee.com)  
**EMEA:** +44 1494 956677 • [EMEAsales@zeevee.com](mailto:EMEAsales@zeevee.com)