zeevee™

Signal Intelligence

ZyPer

Management Platform

User Manual

Updated, October 2018
API Release 1.8x

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 Management Platform includes the Maestro Z web interface. The following browsers are supported:
  - ► Google Chrome version 55.0.2883 or greater
  - ► Mobile Devices: Android tablet 7.0, MS Surface Windows 10, iPad iOS 10.3.2 and higher
- 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

Representatives are available from 9:00 AM to 8:00 PM, Monday through Friday (USA East Coast Time).

Sales

North America:

+1 (347) 851-7364 Phone

sales@zeevee.com

EMEA:

+44 1494 956677 Phone

EMEAsales@zeevee.com

Features

- Pre-configured Linux O/S is maintenance-free and includes upgrades and support.
- Plug & Play operation will discover and enable labeling and control of any number of ZyPer4K, ZyPerUHD or ZyPerHD encoders and decoders.
- 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 5x5 array.
- Create and manage Multi-view displays with up to 9 sources. (ZyPer4K only)

New in Release 1.8

New Features

- Enhanced Security via elimination of access to Linux Shell

See Network Configuration section for updated procedure for setting Server IP Address

New/Updated Commands (See API Command Listing)

- channel
- events
- join hdmi-downmix-audio
- join video-wall
- join window
- redundancy delete down-servers
- revert server
- send CEC
- set decoder power-save
- set server ip
- set video-wall decoder
- set video-wall new-name

New Hardware (See Hardware Specifications)

- Enterprise Grade Rack Mount Server

Contents

1 Getting Started

2 Basic Operation

3 Advanced Operation

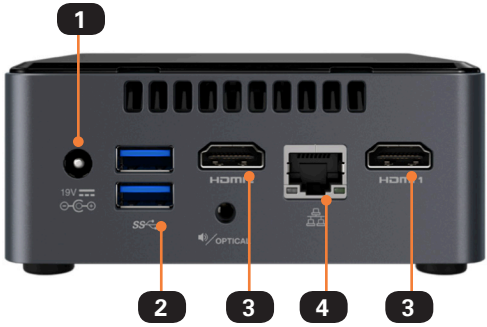
4 Event Mechanism

5 Appendix

1 Getting Started

Panel Descriptions

Rear View



ID	Name	Description
1	Power receptacle	Connect the included power supply to this power receptacle.
2	USB ports	Connect up to two USB 3.0 devices, such as a keyboard and mouse.
3	HDMI	Connect an HDMI cable from this port on the Management Platform to an HD display. (optional)
4	Ethernet port	Connect an Ethernet cable from this port to a switch on the Local Area Network.

Side View

ID	Name	Description
5	Kensington Lock	This port accepts a standard computer laptop locking connector.

Front View

ID	Name	Description
6	USB port	Connect up to two USB 3.0 devices, such as a keyboard and mouse, to these USB ports.
7	Power On/Off	Press this button to manually power-off the Management Platform. When the Management Platform is connected to a power source, it will automatically power-on. In addition, if power is lost or the power is disconnected, the Management Platform will automatically power-on when power is restored.

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.

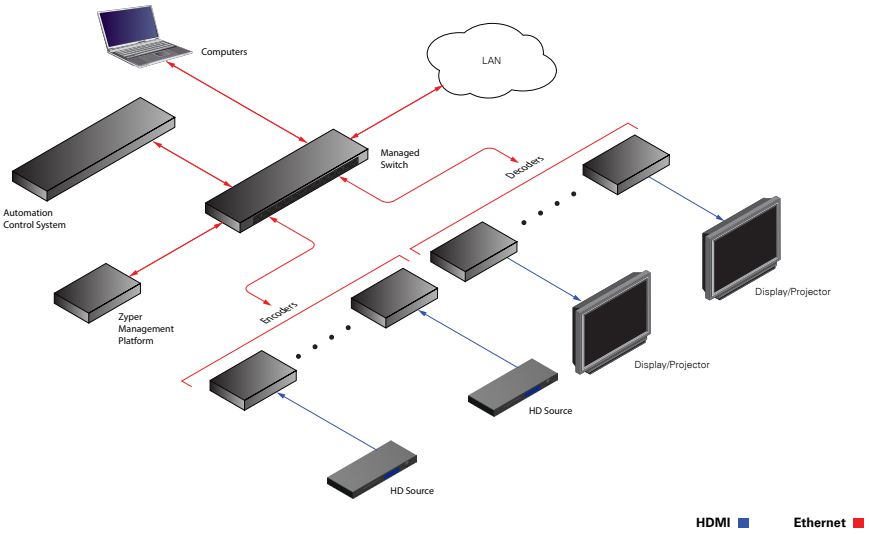
b. Open the Terminal application.

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

d. 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 Maestro Z 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 between the Management Platform and a Display. (Note that a USB keyboard is also required to be connect to the Management Platform)

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

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

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

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

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

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

```
Zyper$ show server info
server(192.168.0.78);
  server.gen; hostname=zyper.local, version=1.7.4.33922, macAddress=1c:1b:0d:82:ff:1a,
  serialNumber=ZZM1H500032B
  server.gen; uptime=0d:2h:4m:6s, freeMem=6.460GB, bootCount=7
  server.gen; runningInVm=false
  server.ip-addressing; dhcp-allocated=true
  server.time; time=Wed Aug  1 21:35:06 2018, timezone=America/New_York
  server.license; productID=031B021C-040D-0582-FF06-1A0700080009, license=none
  server.license; limit=24, knownDevices=5, devicesUp=5, devicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_2.11.5=5

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 Managment Platform

```
Zyper$ show server info
server(192.168.0.78);
  server.gen; hostname=zyper.local, version=1.7.4.33922, macAddress=1c:1b:0d:82:ff:1a,
  serialNumber=ZZM1H500032B
  server.gen; uptime=0d:2h:4m:6s, freeMem=6.460GB, bootCount=7
  server.gen; runningInVm=false
  server.ip-addressing; dhcp-allocated=true
  server.time; time=Wed Aug  1 21:35:06 2018, timezone=America/New_York
  server.license; productID=031B021C-040D-0582-FF06-1A0700080009, license=none
  server.license; limit=24, knownDevices=5, devicesUp=5, devicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_2.11.5=5

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.

```
ZyPer$ 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 63) for a full listing of available commands.

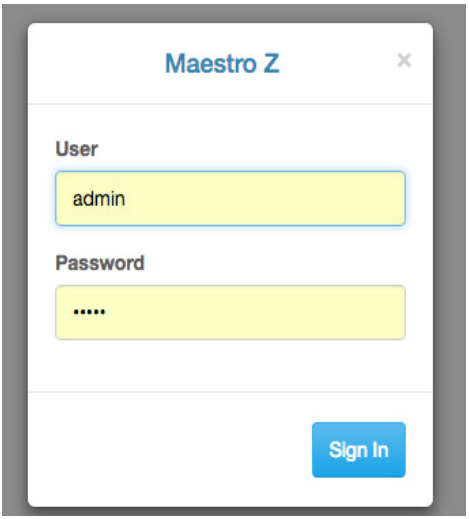
```
Zyper$ set server ip server static 192.168.1.26 255.255.255.0 none
reboot

Success
```

2 Basic Operation

Accessing MaestroZ

1. Open a web browser and enter the IP address of the Management Platform.
2. The login screen for MaestroZ will be displayed.



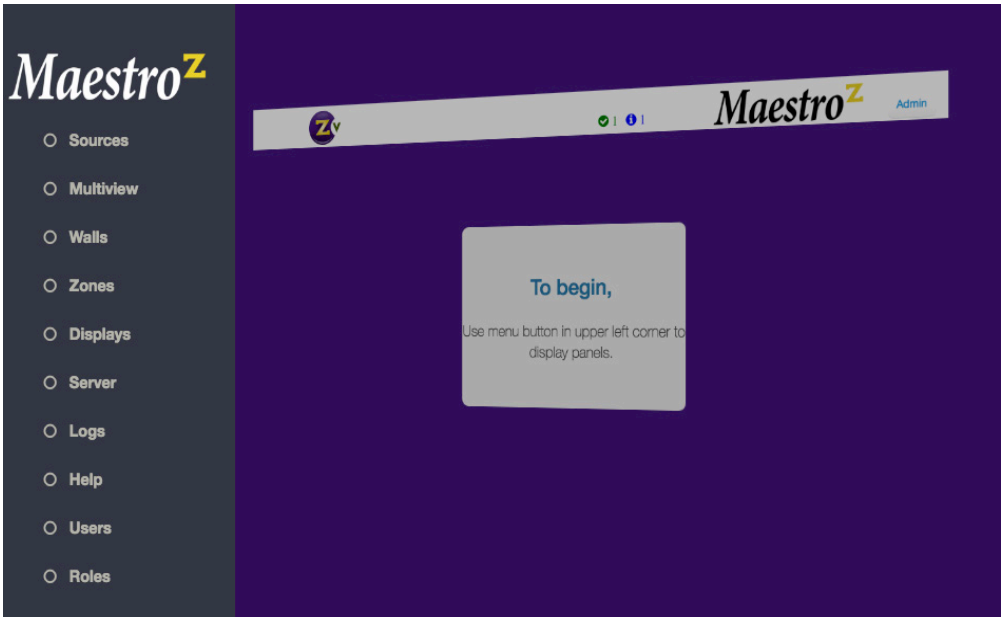
3. Enter the required information in the **Username** and **Password** fields. The default username and password is `admin`. The username and password are case-sensitive. The “admin” password may be changed by the user at any time.
4. Click the **Sign In** button.
5. The **Home** page will be displayed. See the next section for more information.

Home Page

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

1. Login to the Maestro Z. Refer to Accessing MaestroZ (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*, *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.
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.

   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.

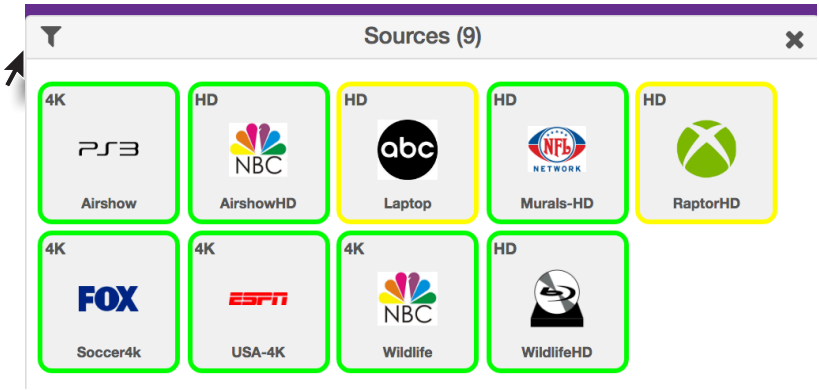
Meaning	Indicator Color
Error	Red

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

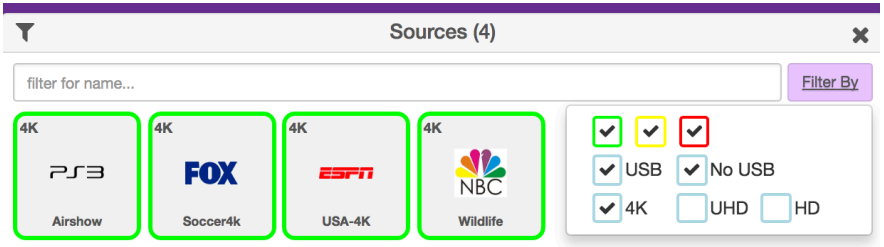
Displaying Devices By Status

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.

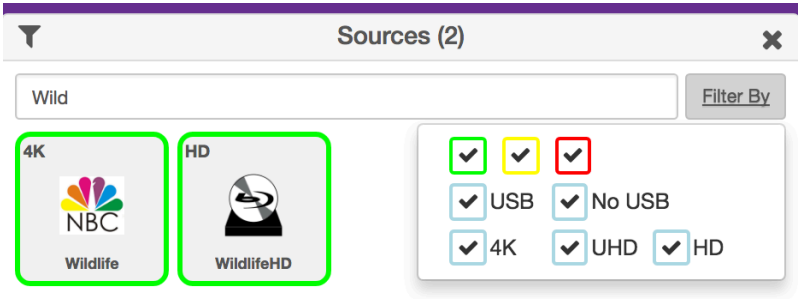


3. Select `every box` from the **Filter** to show all devices.

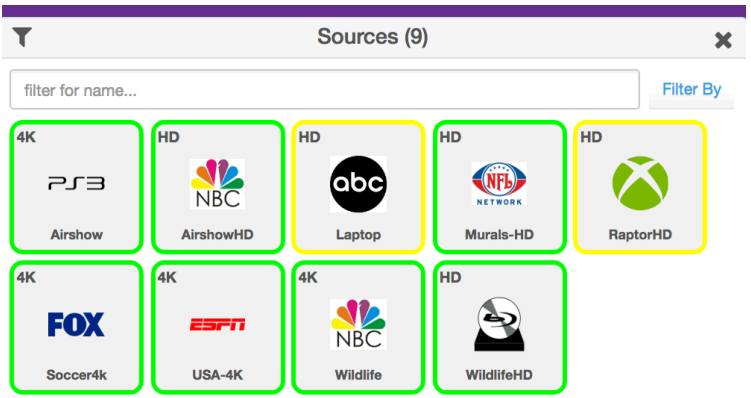
Displaying Devices By Name

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 19).



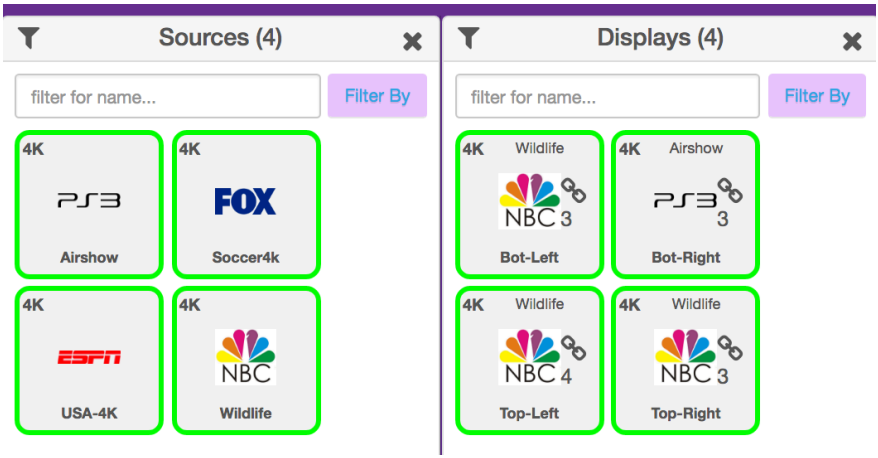
2. Type the desired sequence of characters in the **Filter by name** field to search. In this example, we want to only display the “Wildlife” source devices. To do this, we can enter part of the name, such as “Wild” or even “W” (since no other device name contains the character “W”). All text entries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

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 19).

1. Login to the Maestro Z. Refer to Accessing MaestroZ (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.

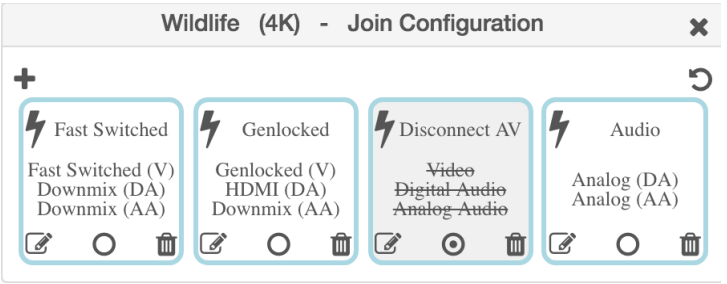
   The join command can also be used to perform the same operation.
   See API Command Listing (page 63) for more information.

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 Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

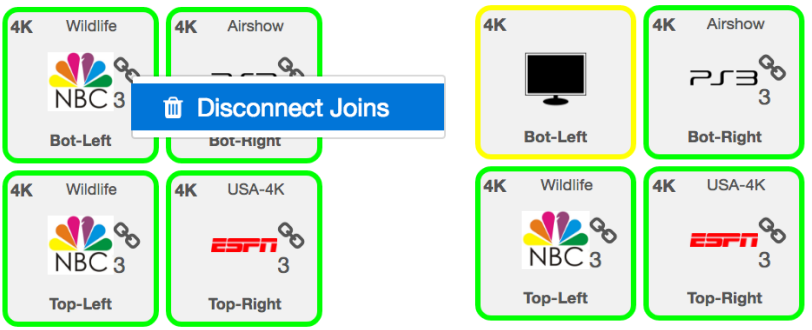
2. On the **Sources** window, right-click on any Source. Select Configure Join and then Disconnect AV.



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)



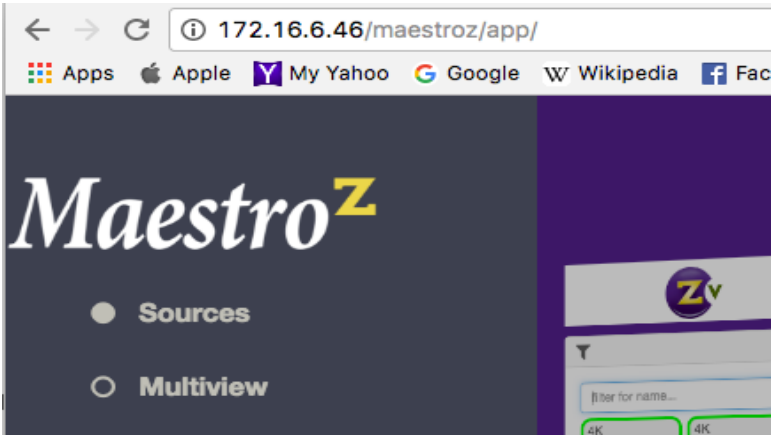
Source Config Page

Configuring Encoders and Decoders

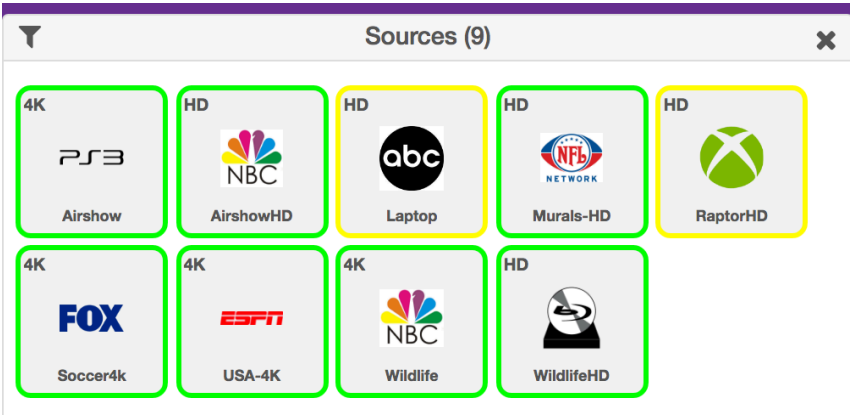
When an encoder or decoder is connected to the network, the Management Platform identifies each unit by its MAC address. This is the default setting. However, when dealing with several units, it is much easier to identify a unit by a string name. It is also possible to assign a preset image to each icon,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

1. Login to the Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

2. Click the **Source or Displays** tab at the left of the page. Joining Encoders to Decoders (page 17)

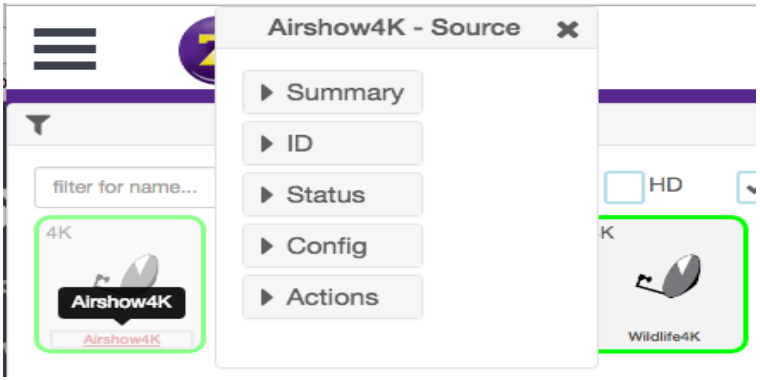


3. All available encoders will be displayed under the **Sources** window.

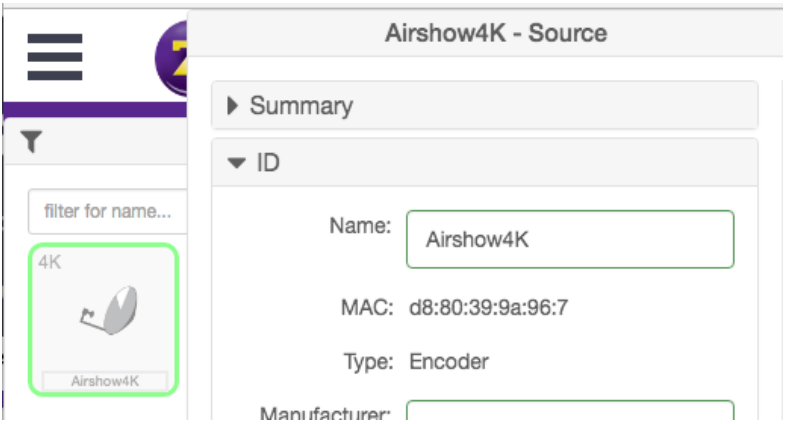


4. Left-click the desired encoder name to display the context menu. In the example below,we will select the encoder named “Airshow4K”.

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



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



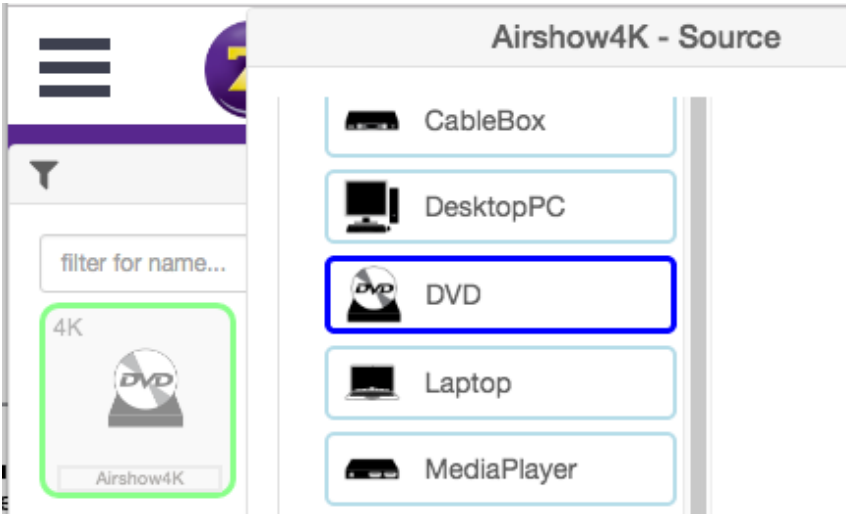
7. Enter a description for the source in the **Name** field. By default, the Management Platform will use the MAC address of the encoder. This field cannot be blank. Names cannot contain spaces. In this example, we will use “Airshow4K”.

   Naming a device can also be done using the `set device general name` command. See API Command Listing (page 63) for more information.

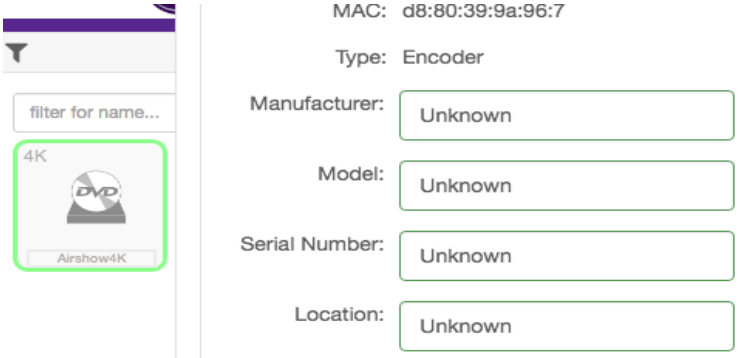
*Each encoder and decoder must have a unique name. Duplicate names are prohibited. It should be noted that assigning a name to an encoder does not actually replace the MAC address. The **Name** field provides a descriptive identifier, rather than a MAC address.*

8. Click the **Config-Icon** drop-down list to select the desired icon for this source. This is optional. If no icon is selected, then the default icon will be used. In this example, we will select **DVD**, since a DVD player 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 63) for more information.



9. Enter the **Manufacturer**, **Model**, and **Serial Number** of the source or display in the appropriate fields. By default, these fields are set to "Unknown". These fields must not be blank. If a custom value is provided, it must *not* contain spaces.



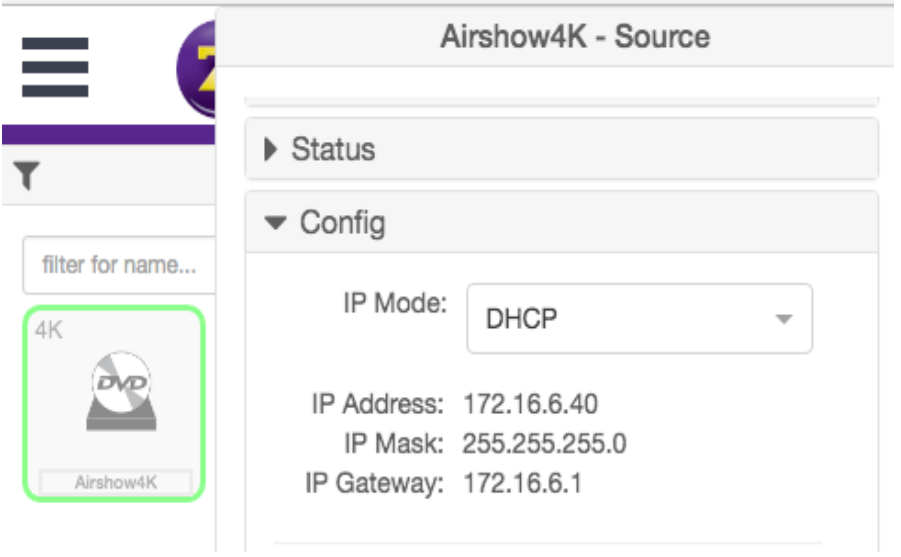
The **Manufacturer**, **Model**, and **Serial Number** can also be assigned using the following commands, respectively. See API Command Listing (page 63) for more information.

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

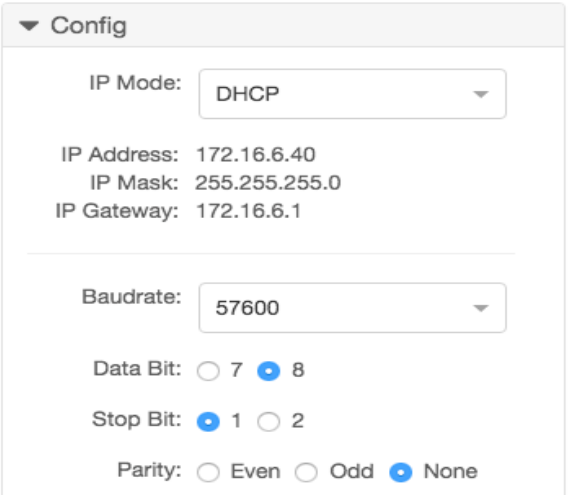
10. Enter the location of the source or display device in the **Location** field. By default, this field is set to “Unknown”. This field must not be blank. If a custom value is provided, it must *not* contain spaces.

11. The `set device source-display location` command can also be used to set these values. See API Command Listing (page 63) for more information for more information.

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



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



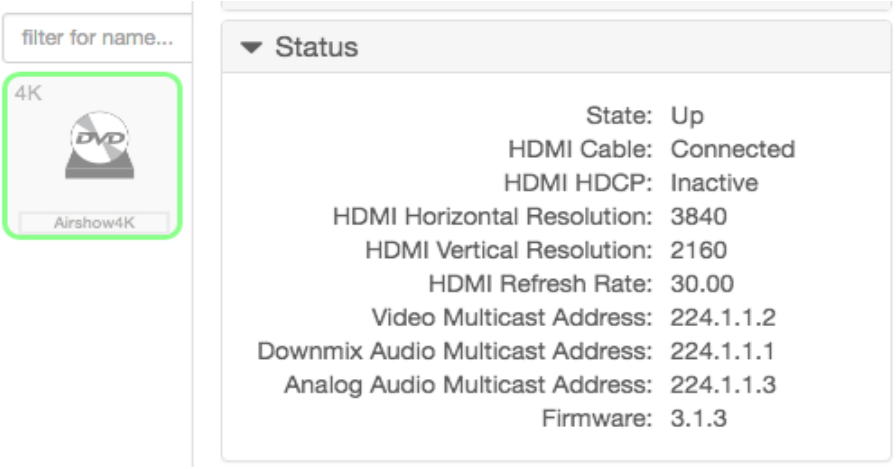
The **Device IP** settings can also be assigned using the `set device ip dhcp` and `set device ip static` commands. When assigning **RS232** settings from the command line, use the `set device rs232` command. See API Command Listing (page 63) for more information.

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 Maestro Z. Refer to Accessing Maestro Z (page 11) for more information.

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

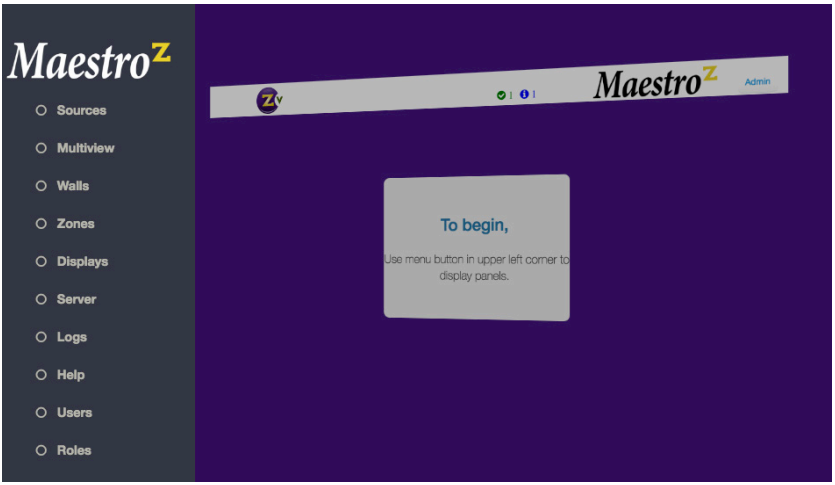


Deleting, Rebooting or Resetting an Encode or Decoder

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

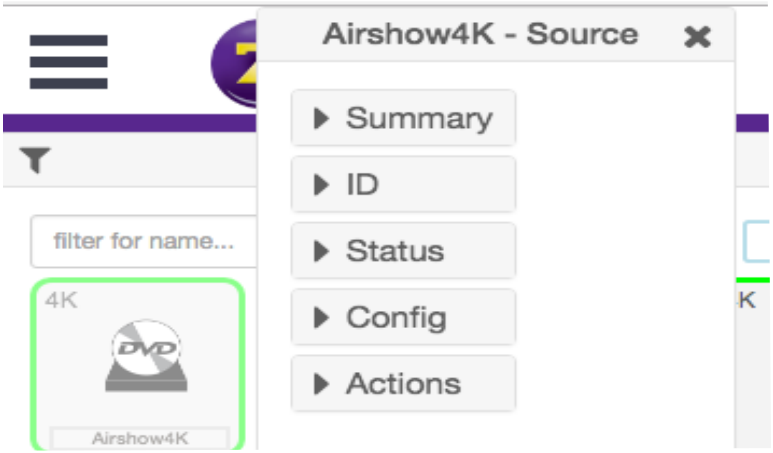
1. Login to the Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

2. Click the **Source** tab at the left of the page.

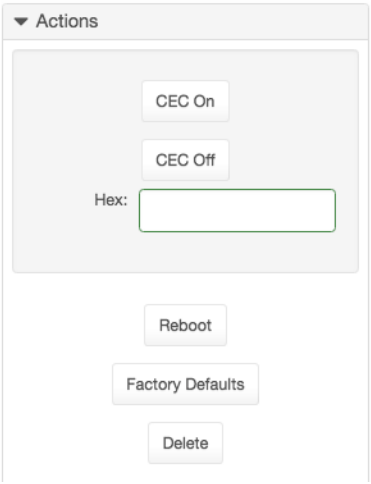


► To delete an encoder or decoder:

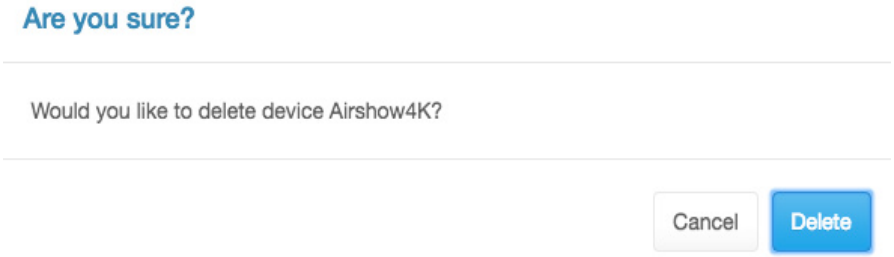
a. Left-click on the desired encoder or decoder and select **Actions** from the context menu.



b. Click the **Delete** button.



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



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

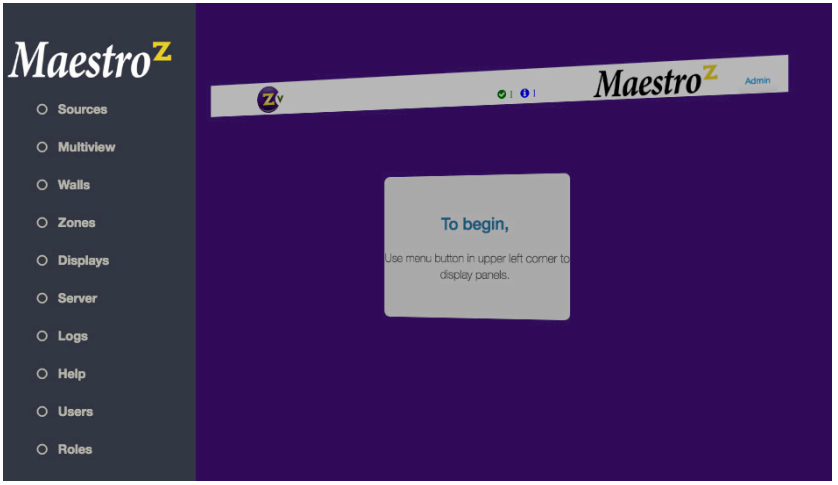
5. This same menu can be used to Reboot the device or set the device back to Factory Defaults by clicking the appropriate button.

Sending a CEC Command

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

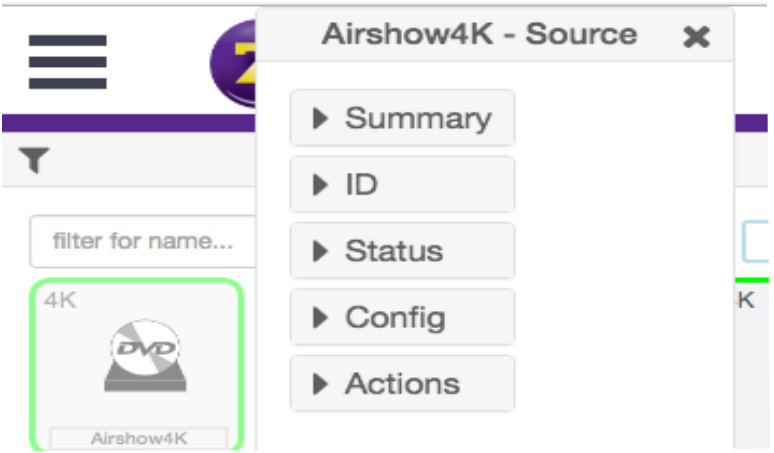
1. Login to the Maestro Z. Refer to Accessing Maestro Z (page 11) for more information.

2. Click the **Source** tab at the left of the page.

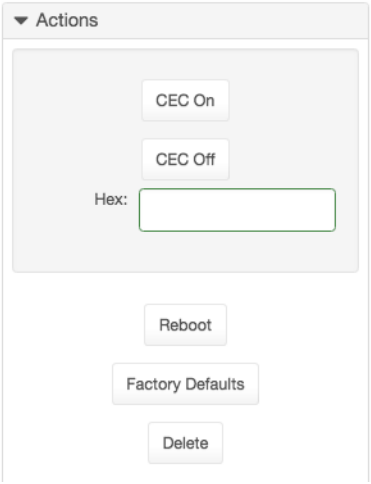


► To send a CEC command:

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



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

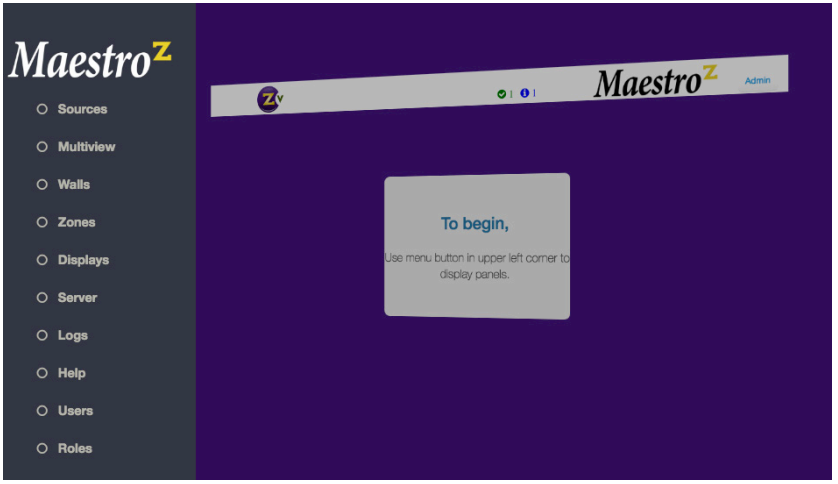


3. The ZyPer4K can also send Hex commands over CEC. Just type the Hex command into the box. When completed, click anywhere outside of the Hex input box to send the command.

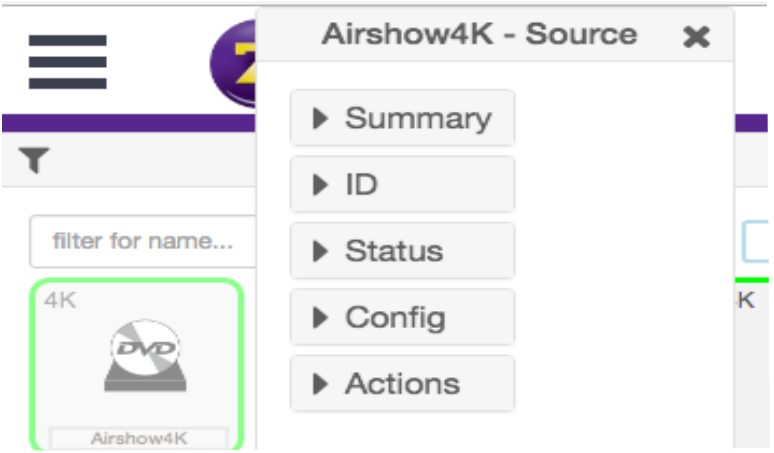
If there is a firmware update available for the ZyPer4K, ZyPerUHD or ZyPerHD, the update can be performed easily from within MaestroZ.

1. Login to the Maestro Z. Refer to Accessing Maestro Z (page 11) for more information.

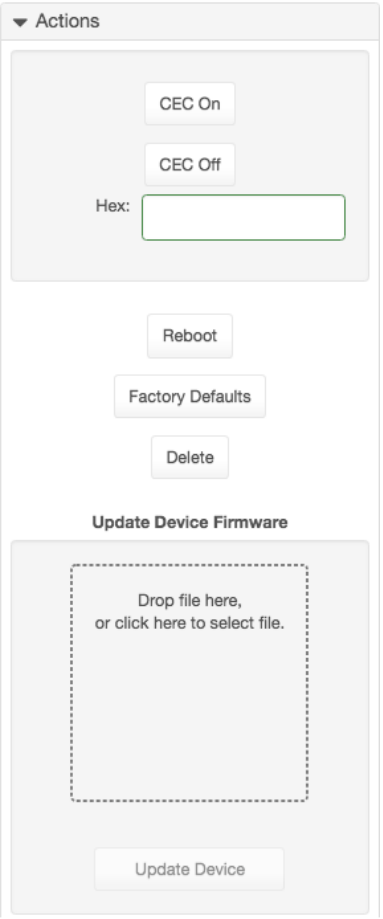
2. Click the **Source** tab at the left of the page.



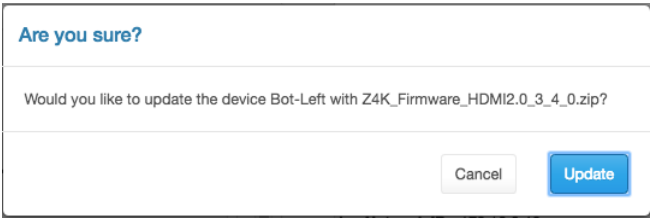
a. Left-click on the desired encoder or decoder and select **Actions** from the context menu.



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



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

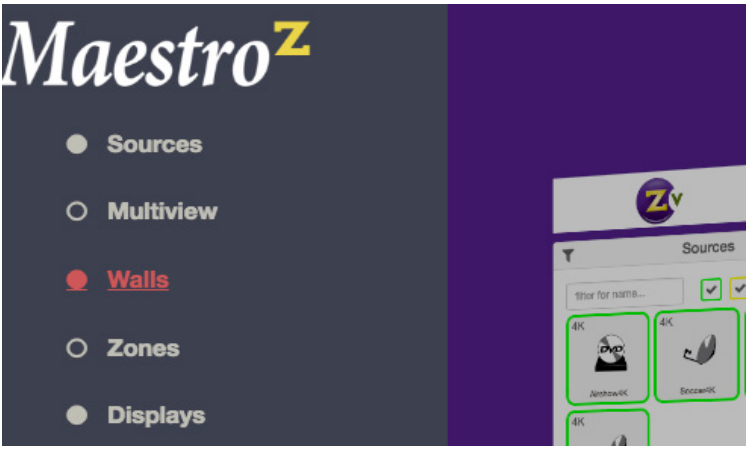


Creating Video Walls

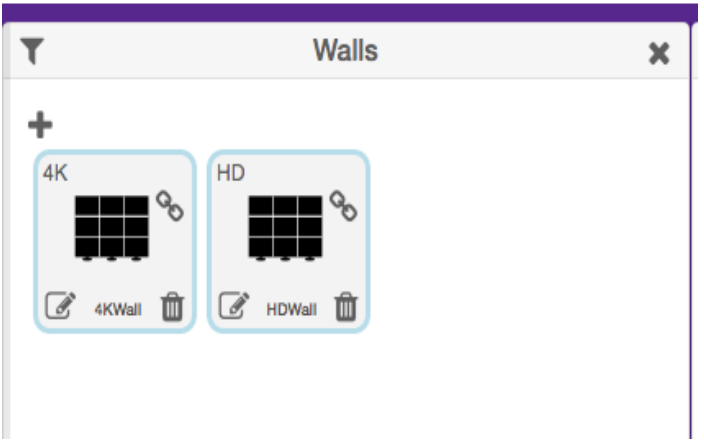
One of the purposes of the Management Platform is to create and manage video walls. A video wall is a collection of displays or projectors arranged in a square or rectangular fashion. The source is then “mapped” to each display, thereby creating one large display from multiple, smaller displays.

1. Login to the Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

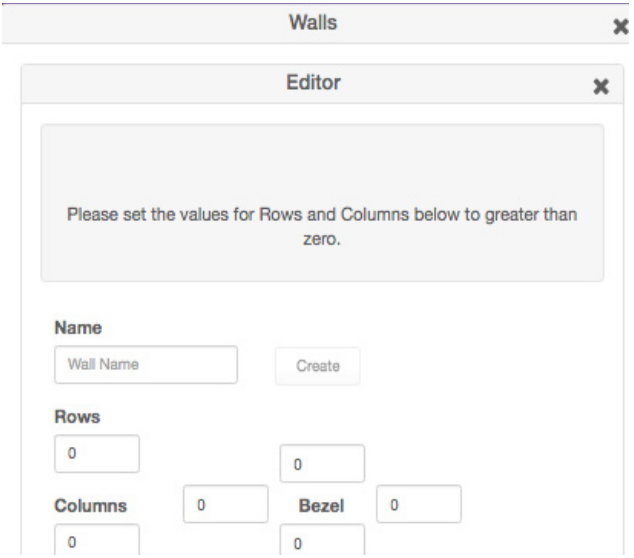
2. Click the **Walls** tab at the left of the page.



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



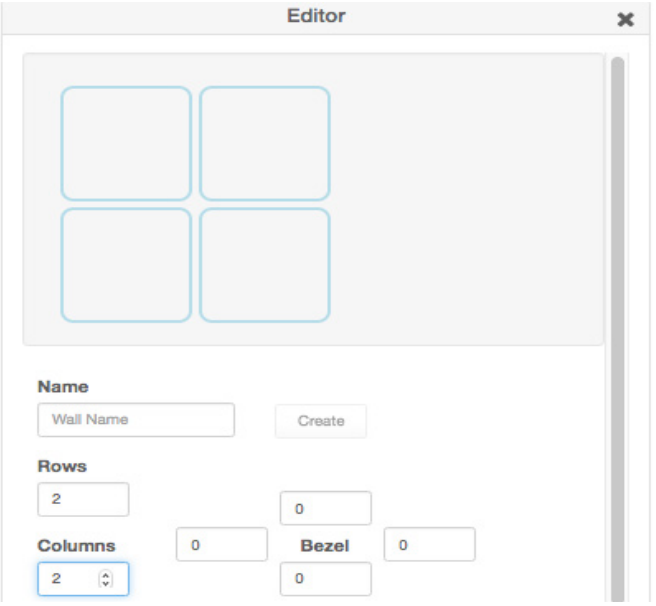
4. This will bring up the wall **Editor.**



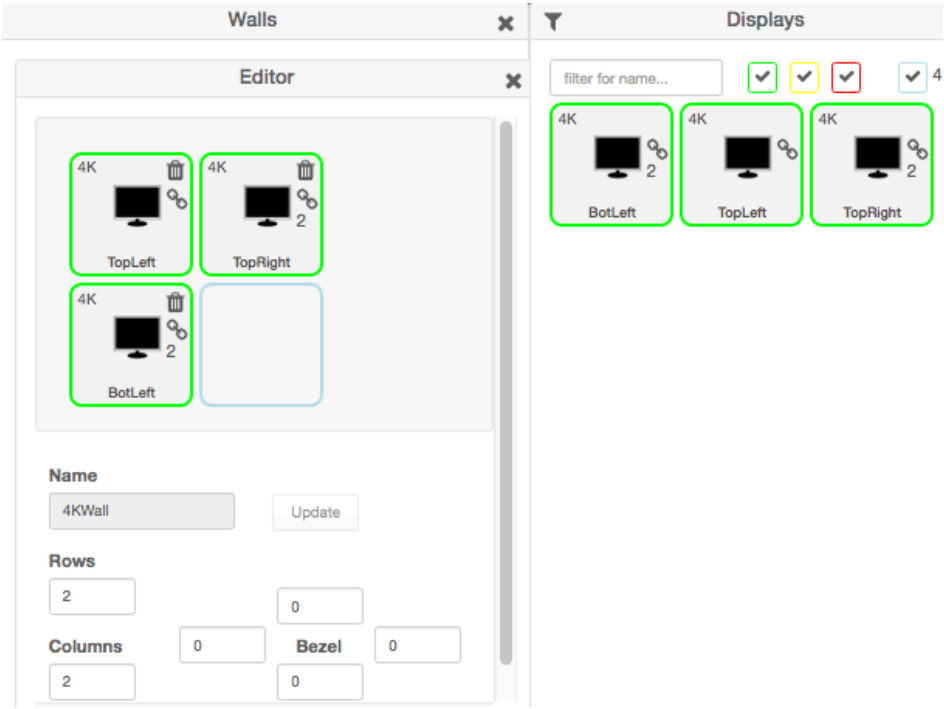
5. Enter the number of display columns in the **Number of Columns** field.

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

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



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

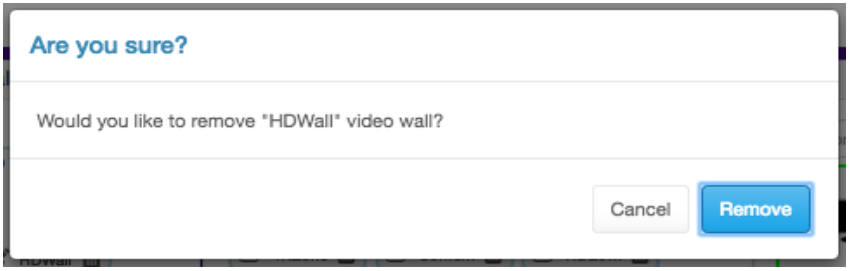


8. Leave the **Bezel Top**, **Bezel Bottom**, **Bezel Left**, and **Bezel Right** set to 0. Bezel compensation will be covered in the next section.

9. Click the **Create** button.

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

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



Bezel Compensation

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

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

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

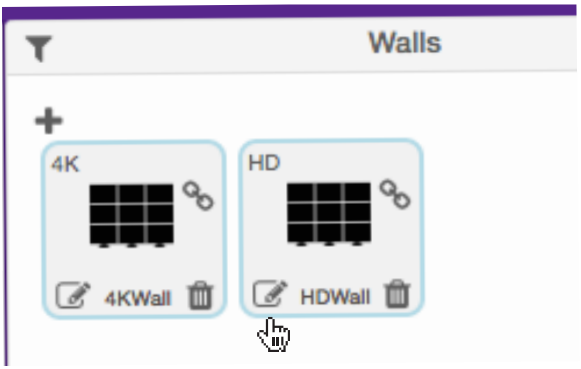
Without Bezel Compensation

With Bezel Compensation

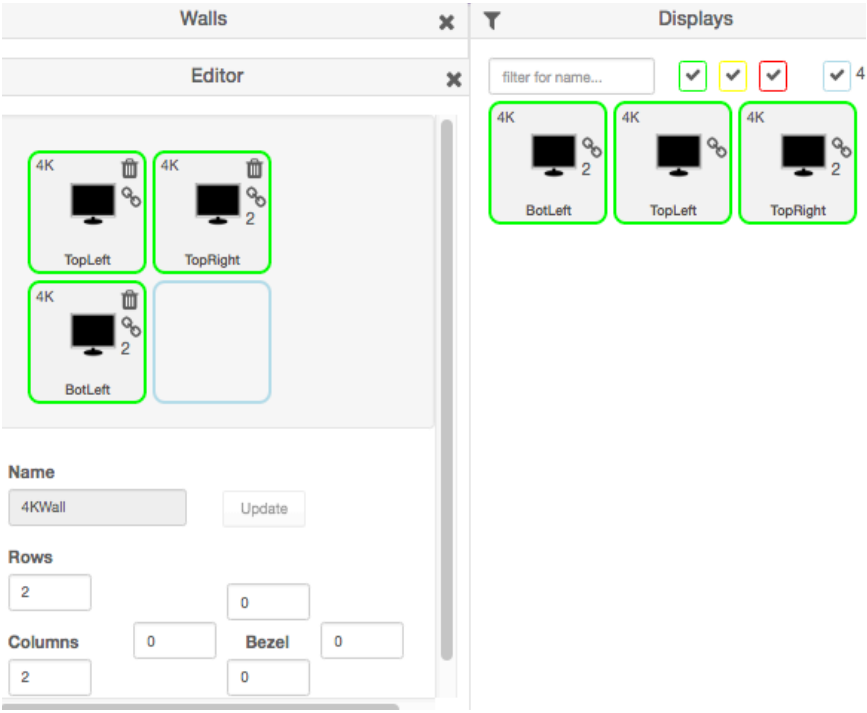
   It is recommended that when the video wall is set up for the first time, that these values be set to zero. *Bezel compensation* can be changed at any time.

> *Bezel compensation is always measured in pixels.*

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



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



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

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

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

7. **Note:** Only the ZyPer4K allows bezel adjustment. Video walls with the ZyPerHD or ZyPerUHD do not allow bezel adjustment.

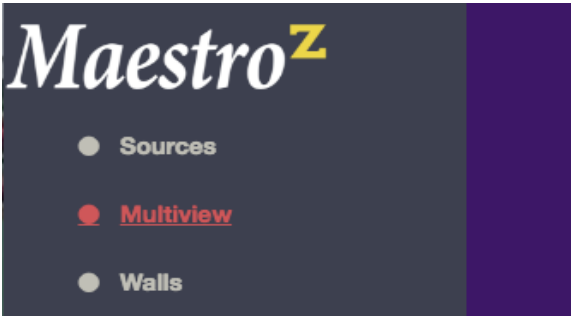
Creating a Multiview Screen

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 9 sources displayed on a single display in a variety of preset patterns.

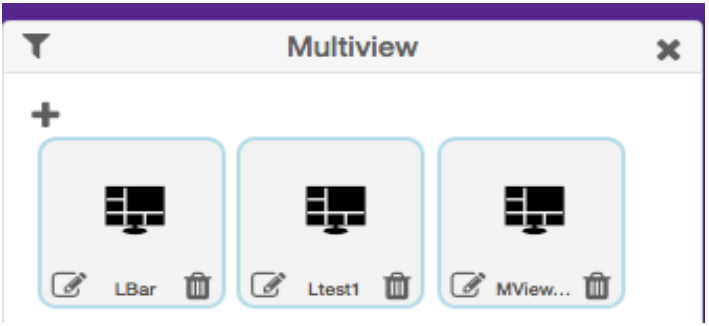
Note: Only the ZyPer4K versions with the HDMI 2.0 feature support multiview screens. This feature is not supported by the ZyPerHD, ZyPerUHD or ZyPer4K units with only HDMI 1.4 support.

1. Login to the Maestro Z. Refer to Accessing Maestro Z (page 11) for more information.

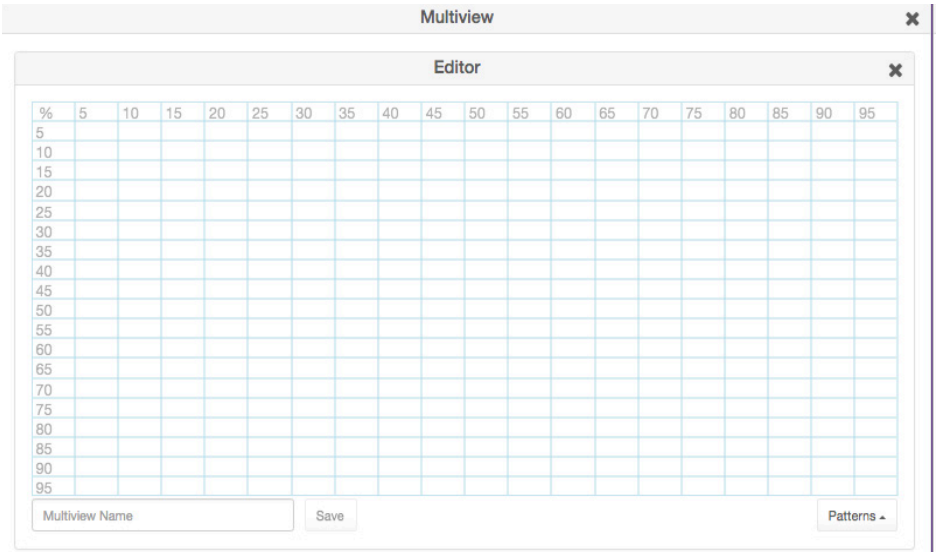
2. Click the **Multiview** tab at the left of the page.



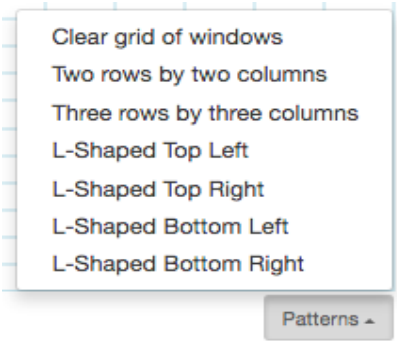
3. In the **Multiview** pane, click on the **+** button to create a new video wall.



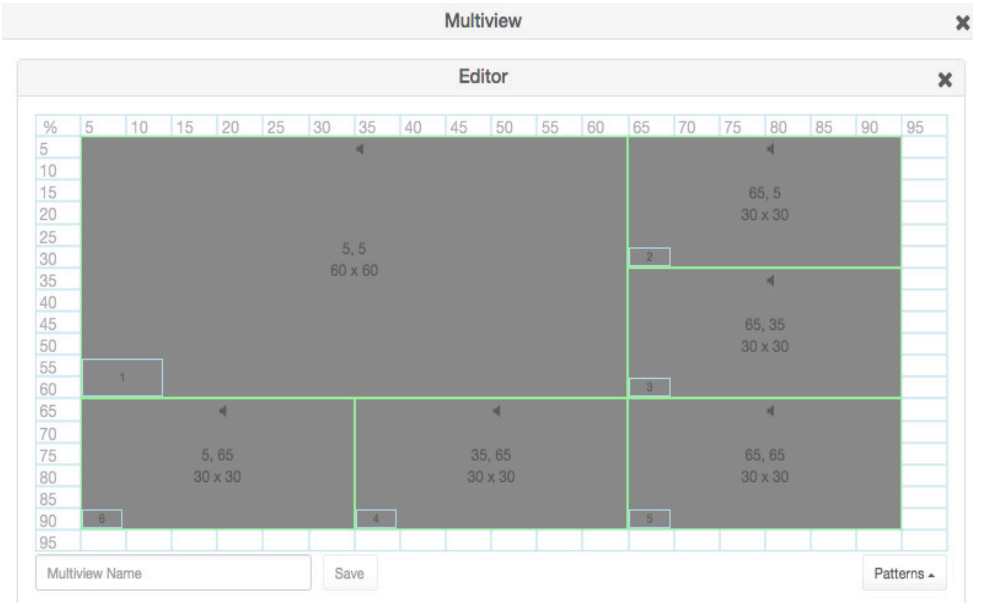
4. This will bring up the Multiview editor where you can create and customize a multiview display.



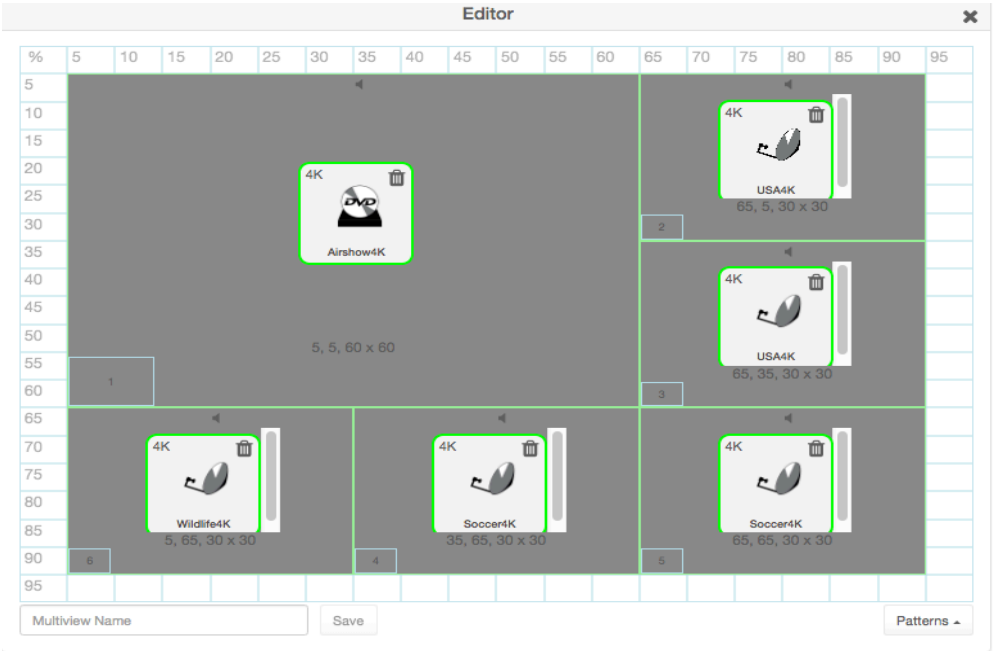
5. Click on the **Patterns** button in the lower right corner to bring up a list of pre-defined patterns.



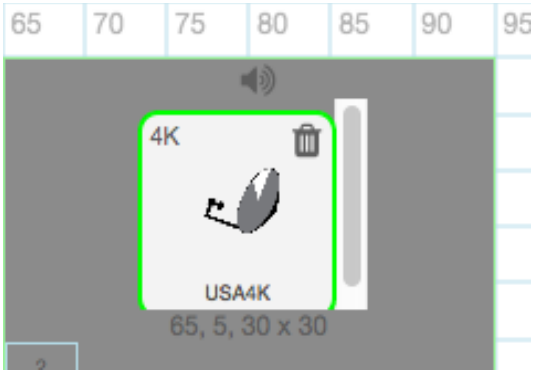
6. After selecting a pattern you will see the grid fill in with the places to drag sources.



7. Drag sources into the various locations. Note: The same source can be dragged into muliple windows as long as the windows are the same size. Dragging a source onto differnent 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.

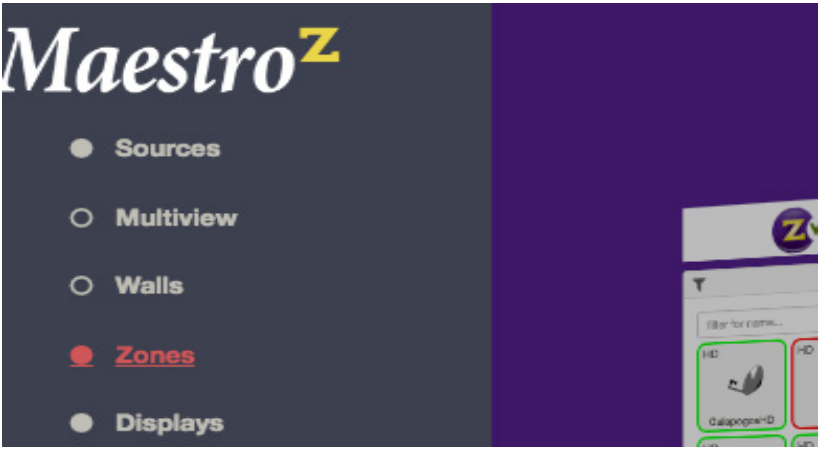
10. To use the Multiview, drag the newly created mulitview onto a Display in the **Display** pane. Note that the Display must be a 4K compatible display for this funcitonality to work.

Creating Zones

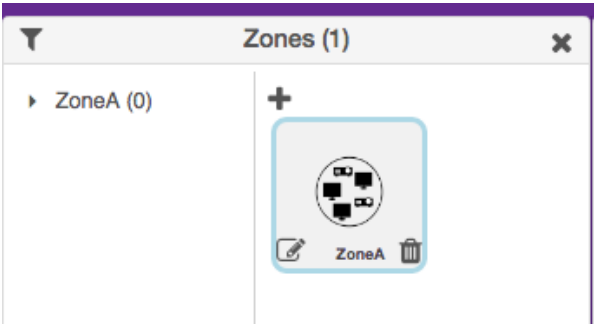
In many installations, displays are placed in more than one room. These rooms are often referred to as *zones*. Creating a *Zone*, using the Management Platform, allows you to organize these displays in a group. *Video Walls* can also be added to *Zones*.

1. Login to the Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

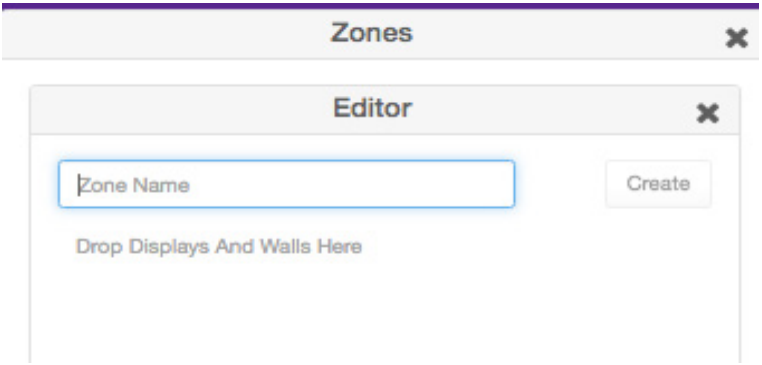
2. Click the **Zones** option at the left of the page.



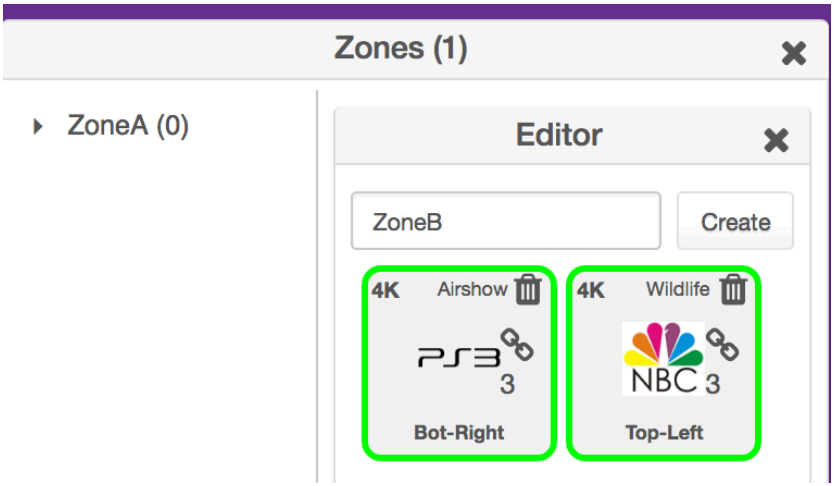
3. In the Zones window, click on the + symbol to add a Zone.



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



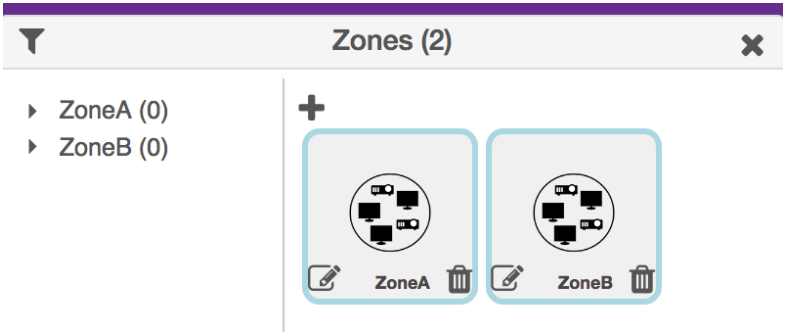
4. Continue adding the desired displays (or *video walls*) to the drop-pane.



5. 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*.

7. Click the **Create** button to save the *Zone*. Close the editor window to exit without saving changes.

8. The new *Zone* will appear under the **Zones** window.



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

Server Functions Page

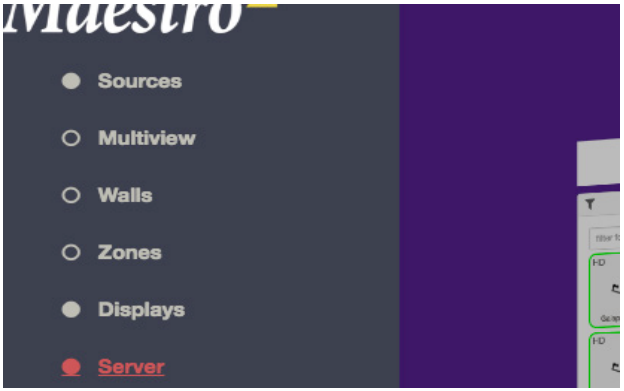
The majority of this page is used to display general information about the Management Platform. In addition, this page provides controls for setting the EDID mode and rebooting or restarting the server. Server firmware can also be updated from this window.

Server Information

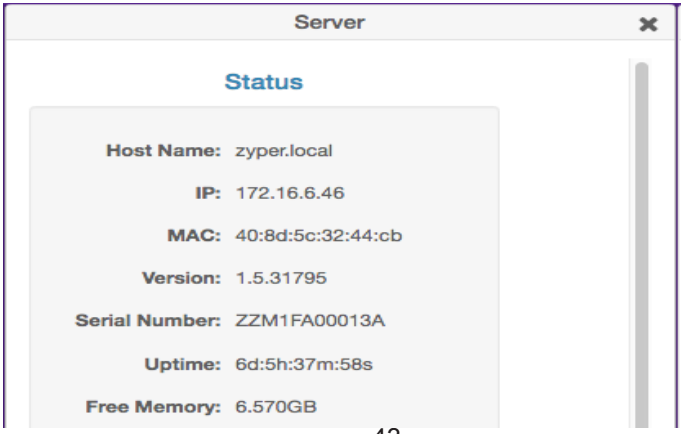
This section provides information about the Management Platform, such as the host name, IP address, MAC address, version, serial number, uptime duration, and free memory.

1. Login to the Maestro Z. Refer to Accessing MaestroZ (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.



Setting the EDID Mode

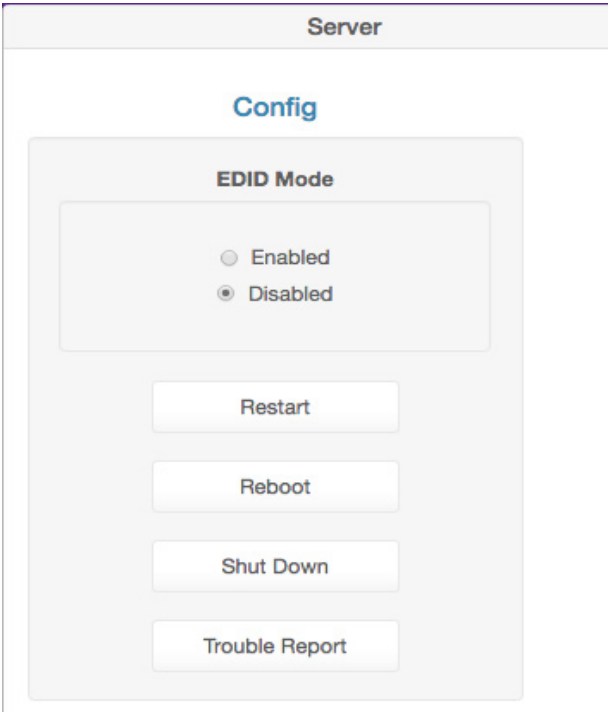
By default, Auto EDID mode is *enabled*. This means that the Management Platform will compare the encoder EDID with the decoder EDID. If they are different, then the EDID from the decoder (sink) will be used by the encoder (source). Setting the EDID Mode affects all join modes: fast-switched, genlocked, and video-wall. Refer to the `join` command in the API Command Listing (page 63) section for more information.

1. Login to the Maestro Z. Refer to Accessing MaestroZ (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.

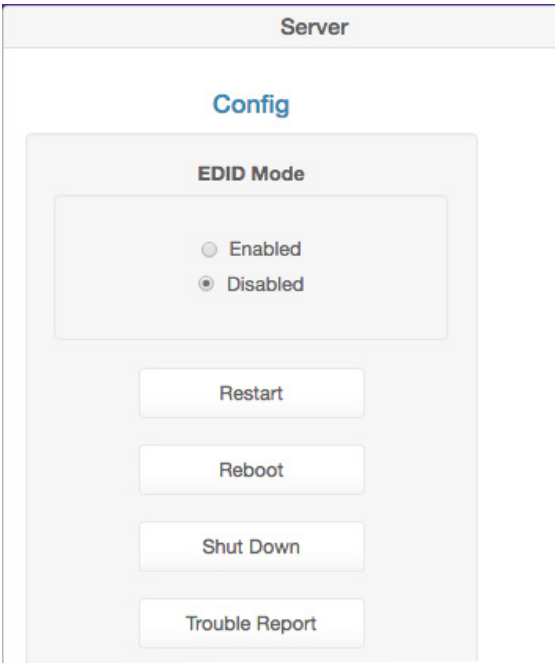
Note: There is a check box at the bottom of the Server window for "Show advanced controls" This box must be checked to enabled changing of the EDID mode setting.



Server Reboot, Restart, and Shutdown

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

1. Login to the Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.
2. Click the **Server** tab at the left of the page.
3. Reboot, Restart and Shutdown options will appear in the **Server** window under the Status information. You may need to scroll down in the window to see these options.



4. Click the desired button under **Server Functions**.

   ► **Restart**
   Restarts the server. Linux is not restarted.

   ► **Reboot**
   Linux is rebooted.

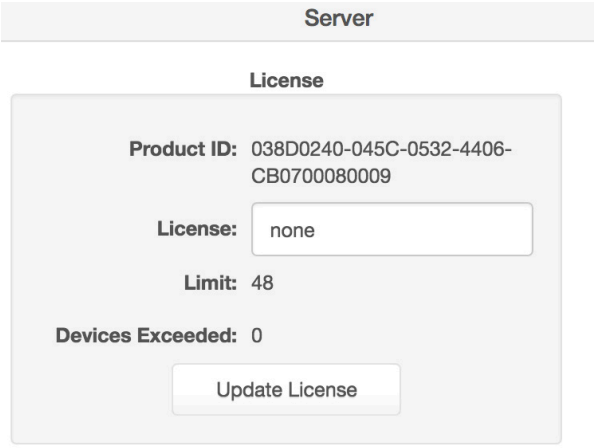
   ► **Shut Down**
   Shuts down the Management Server.

   ► **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 Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.
2. Click the **Server** tab at the left of the page.
3. Scroll down to the License option..

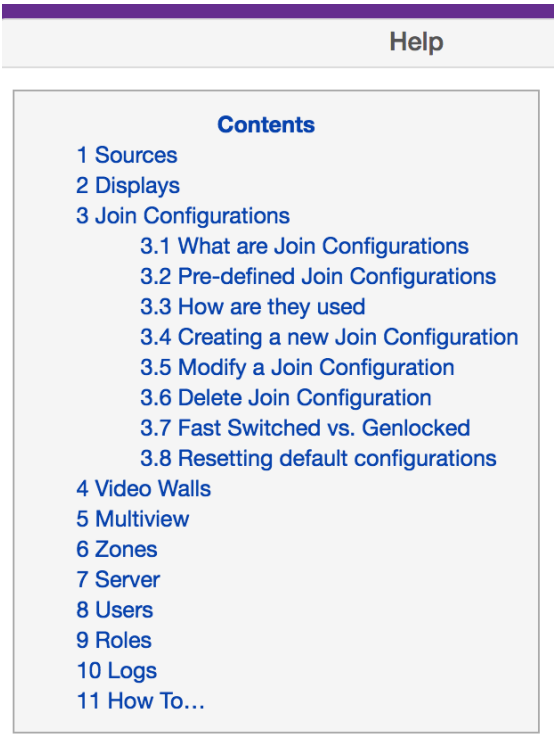


4. Record the Product ID number and provide this to ZeeVee. This ID is used by ZeeVee to generate the new license key when purchased by a customer.
5. Once received, you can enter a new license key as provided by ZeeVee to increase the limit on the number of endpoints. (Please contact ZeeVee support for additional information)
6. Please see the Appendix of this document for information on updating the Server Software. **Note:** In a redundant environment, Software must be updated on slave first, then switchover, then update on new slave. See online help section 11.2

Help Page

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

1. Login to the Maestro Z. Refer to Accessing MaestroZ (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.



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

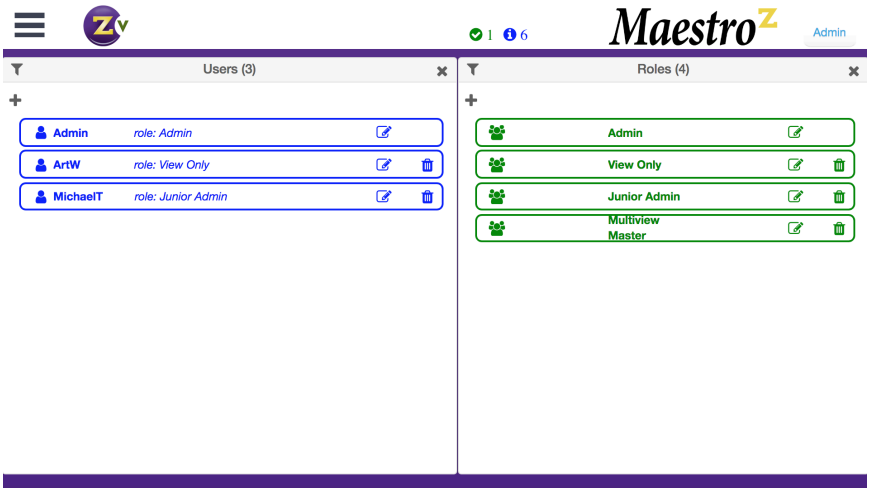
Users and Roles Pages

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 Maestro Z. Refer to Accessing MaestroZ (page 11) for more information.

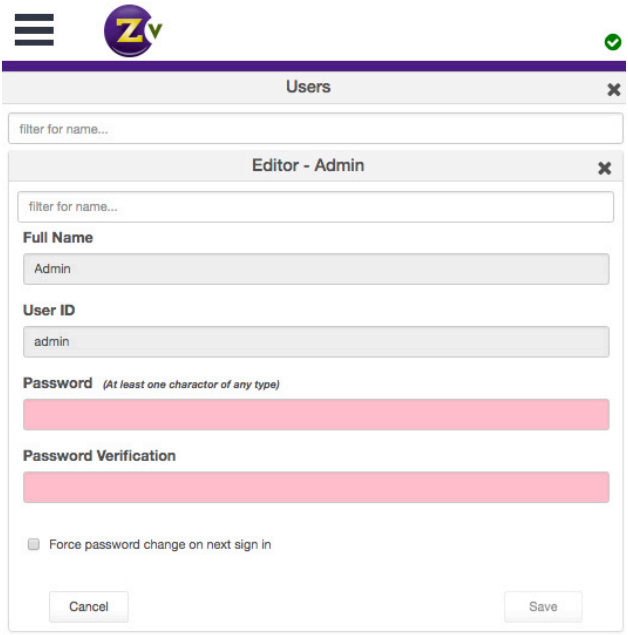
2. Click the **Users** tab and **Roles** tab at the left of the page.

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.



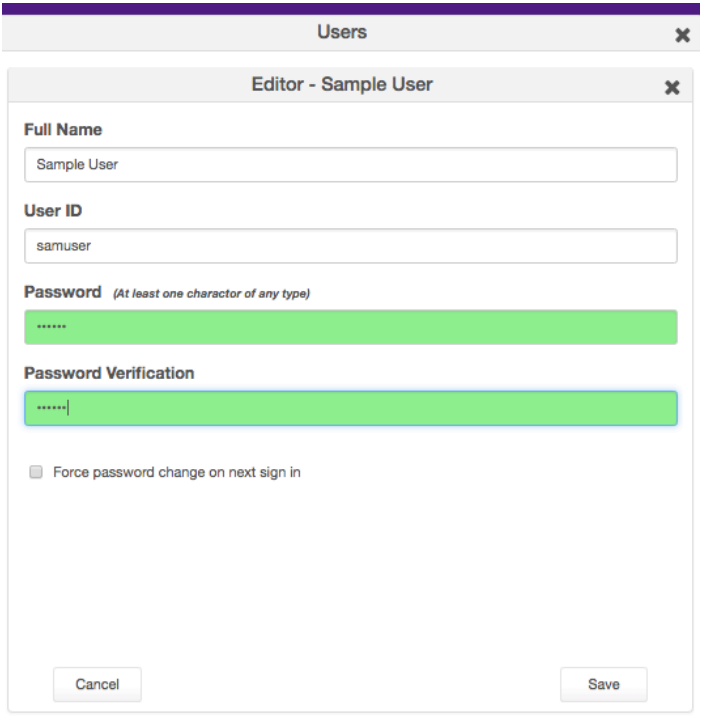
Enter the new password on both the “Password” and “Password Validation” lines and then hit the “Save” button in the lower right corner.

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

Adding a New User

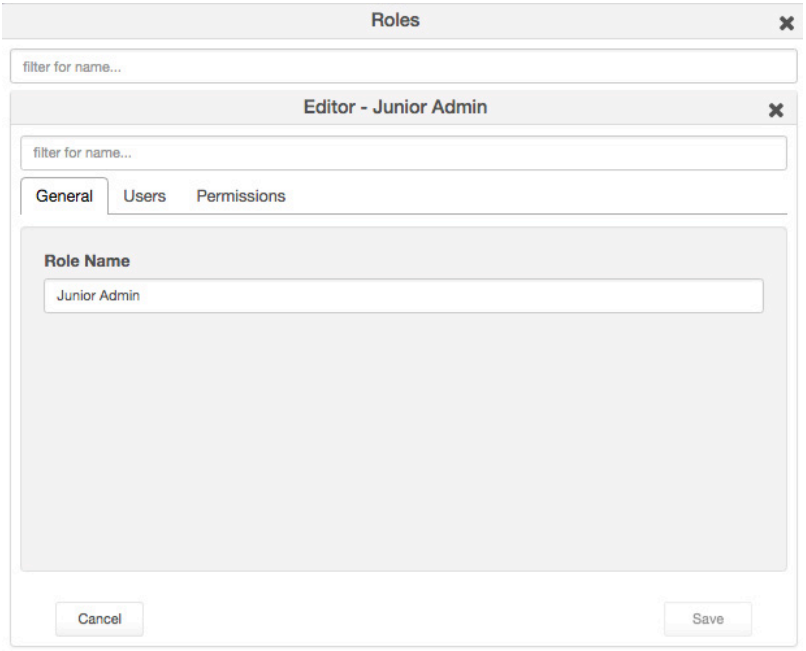
To create a new user click on the "+" symbol in the **Users** window. The screen shown below will appear. Enter the Full Name, User ID and Password for the new user. You can keep this password or force the user to generate a new password the first time they login by clicking the "Force password change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

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



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 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 MaestroZ

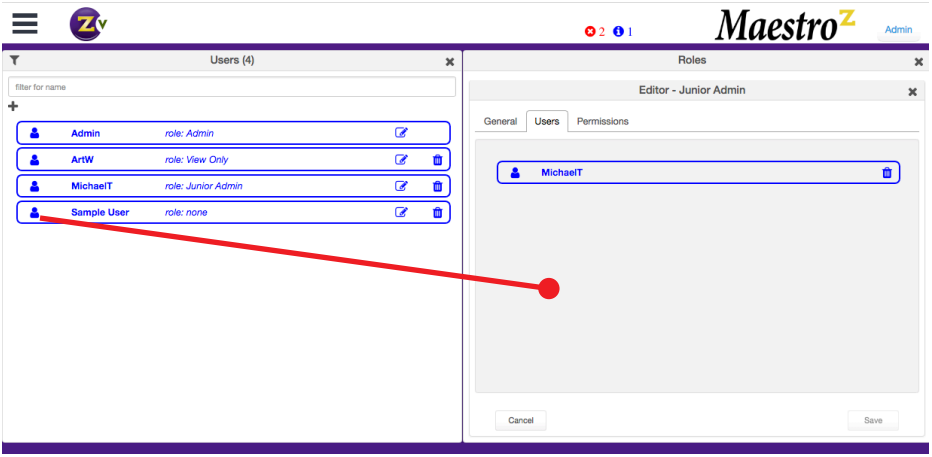
Config: User is allowed to alter or create new configurations within the designated tab. For example the user can create a new Wall, Multiview or Zone.

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

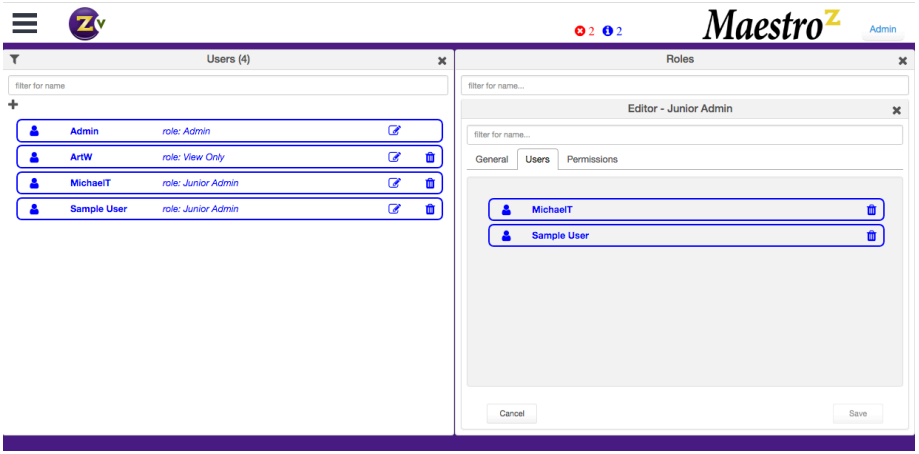
Click on the appropriate levels of access to set permission. The process is color coded to make it easier. Green = Full access, Yellow / Orange = Partial access, Red = No access.

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, “Sample User” 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

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

```
Zyper$ help
show server info
show server config
set server timezone <posix-timezone-name:string>
set server auto-edid-mode enabled|disabled
set server api-password [<password:string>]
...
...
...
script <file-name> [loop]
sleep <millisecond-sleep:int>
Zyper$
```

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 <encoder:macOrNameOrNone> <decoder:macOrName> genlocked
join <encoder:macOrNameOrNone> <decoder:macOrName> fast-switched
join <encoder:macOrName> <decoder:macOrName> analog-audio
Zyper$
```

Setting the Time Zone

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

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.*

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

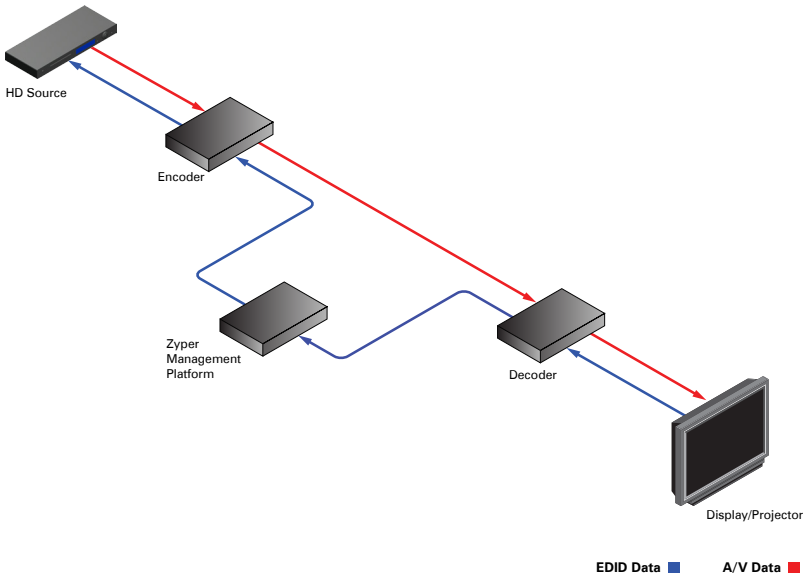
Use the `show server info` command to verify the correct time zone has been set.

```
Zyper$ show server info
server(172.16.6.46);
  server.gen; hostname=zyper.local, version=1.7.1.33348,
 macAddress=40:8d:5c:32:44:cb, serialNumber=ZZM1FA00013A
  server.gen; uptime=0d:19h:20m:23s, freeMem=6.406GB, bootCount=116
  server.gen; runningInVm=false
  server.ip-addressing; dhcp-allocated=true
  server.time; time=Fri Dec  8 09:57:27 2017, timezone=America/
New_York
  server.license; productID=038D0240-045C-0532-4406-CB0700080009,
 license=none
  server.license; limit=48, knownDevices=19, devicesUp=19, devices
Exceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_2.11.3=8, num_2.11.4=3,
 num_3.2.2=8
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 63) 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] [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-edit BlueRay1 myEDID
   ```

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

   ```
   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

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

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

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

Here, we handle the AJAX response:

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

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

The JavaScript object that is returned is:

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

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

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

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

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 blankinig 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 automaticaly reduced to 8-bits at output	HDR input is maintained exactly as input at the output
Dolby / Encoded Audio	Dolby or other encoded audio formas are not passed from encoder to decoder	Dolby or other encoded audio formats are passed from encoder to decoder
Video Wall	Video walls are not supported in Fast-Switched mode. (Join command for walls defaults to Genlock-scaled)	Video walls are technally always in Genlock-scaled mode
Multiview	Multiview is supported in Fast-Switch mode	Multiview is not supported in Genlocked mode
USB, IR, RS232	None of these items are associated with Fast-Switched or Genlocked mode	

API Command Listing

Command	Description
channel	Cycles up or down through encoders. Used to change channels.
create multiview	Creates a new multiview display (ZyPer4K only)
create video-wall	Creates an empty 2x2 video wall.
delete all-configuration	Deletes all encoder/decoder and server information from the Management Platform
delete device	Deletes the specified encoder or decoder from the Management Platform database.
delete multiview	Deletes the specified multiview from the Management Platform database. (ZyPer4K only)
delete multiview-window	Deletes a specific window from an existing multiview (ZyPer4K only)
delete video-wall	Deletes the specified video wall from the Management Platform database.
events	Causes the event mode to be entered
factory-defaults device	Sets the specified encoder/decoder to factory-default settings.
flash-leds	Plysically identifies the specified encoder/ decoder on the network using LED flashes. (ZyPer4K only)
join	Switches audo and/or video from source to display or video wall
join video-source	Selects audio feed to follow a video join
load encoder-edid	Uploads an EDID file to the specified encoder
redundancy switchover	Swap Management Platform Master and Slave
redundancy delete down-servers	Removes no longer present servers from list of redundent servers
restart device	Restarts the specified encoder/decoder
restart server	Restarts the Management Platform
revert server	Switch to a previously installed version of the API

Command	Description
`save device-edid`	Saves the EDID from a decoder to a local file
`send`	Sends an IR, CEC or RS232 string to the specified device
`script`	Executes the specified AJAX/JSON or text script.
`send`	Sends an IR or RS232 string to the specified device
`set encoder analog-audio-out source`	Sets the source of Analog audio output for specified encoder (ZyPer4K and ZyPerUHD only)
`set encoder hdcp-mode`	Sets the HDCP compatibility at the encoder side
`set decoder display-mode`	Sets default decoder output to crop, stretch or box
`set decoder display-size`	Set decoder output size to auto or manual resolution. (Width, Height, FPS)
`set decoder analog-audio-out source`	Sets the source of Analog audio output for specified decoder
`set decoder edid-prefer-mode`	Sets the preferred resolution from the display EDID
`set decoder hdmi-audio-out source`	Sets the source of HDMI audio output for specified decoder
`set decoder power-save`	Enables or disables power-save feature (ZyPerUHD only)
`set device ethernet-management port`	Enables or disables the 1G Ethernet utility port for the specified device (ZyPer4K only)
`set device general name`	Sets the name for the specified device.
`set device ip dhcp link-local`	Sets the specified device to DHCP or Link-Local mode (ZyPer4K only)
`set device ip static`	Sets the specified device to static mode (ZyPer4K only)
`set device rs232`	Sets the RS232 settings for the specified device
`set device send-ip-mcast-range`	Sets allowable range of multicast addresses for selected devices (ZyPer4K only)
`set device source-display iconImageName`	Sets the icon image for the specified device.

Command	Description
`set device source-display location`	Sets the location name for the specified device.
`set device source-display manufacturer`	Sets the manufacturer name for the specified device.
`set device source-display model`	Sets the model name for the specified device
`set device source-display serialNumber`	Sets the serial number name for the specified device
`set device usb-filter`	Allows restrictions to USB use on selected device (ZyPer4K only)
`set device video-port`	Selects active input port for ZyPer4K units with multiple inputs (ZyPer4K only)
`set multiview`	Assigns source to a position and size within a multiview display (ZyPer4K only)
`set multiview audio-source window-number`	Selects the input source to provide Audio for multiview display (ZyPer4K only)
`set responses rs232-term-chars`	Specifies the RS232 termination string
`set server auto-edid-mode`	Sets the EDID mode
`set server api-password`	Sets the API password
`set server enhanced-logging`	Enabled/disables enhanced event logging feature (ZyPer4K only)
`set server ip`	Sets the IP address of the Management Platform
`set server license`	Sets server license. (Max endpoints)
`set server redundancy`	Set a virtual IP address/mask for Master and Slave Management Platforms
`set server timezone`	Sets the time zone
`set video-wall`	Creates a new video wall or modifies an existing wall
`set video-wall decoder`	Assigns the specified decoder to a position within the video wall

Command	Description
`show device capabilities`	Shows detailed capabilities of specified device or devices
`show device config`	Shows detailed configuration information for specified device or devices
`show device connections`	Shows encoder connections to decoders
`show device status`	Provides detailed status informatoin for specified device or devices
`show multiviews config`	Lists all created multiviews with source, position and size info (ZyPer4K only)
`show multiviews status`	Lists all created multiviews with source, datarate and multicast address info (ZyPer4K only)
`show responses`	Displays the lastChangeId for the specified device
`show server config`	Displays the IP address and EDID mode of the Management Platform
`show server info`	Displays Management Platform information
`show server redundancy`	Displays information about Master and Slave Management Platforms
`show video-walls`	Displays a list of all created video walls
`shutdown server`	Shuts down or reboots the Management Platform
`sleep`	Sets a time dely, in milliseconds
`stop encoder`	Stop a specified stream (ZyPer4K only)
`start encoder`	Start a specified stream (ZyPer4K only)
`switch`	Switches IR or RS-232 between devices
`trouble-report`	Generates a trouble report
`update device`	Updates the individual encoder or decoder unit
`update server`	Updates the Management Platform software. See Updating the Software (page 167) for more information.

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 fast-switch connection types is supported. If there was already a connection of some other type, it is changed to Fast-Switched.

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 Fast-Switch mode the join video-source <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 video-source`

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 Maestro Z.

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 36) for information on managing multiview displays in the built-in Maestro Z.

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 video-wallmultiview
delete multiview-window
set video-wallmultiview
set video-wall-decodermultiview audio-source window number
show multiviews config
show multiviews status
```

create video-wall

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

Use the `set video-wall-encoder` 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 video-wall` command.

Refer to Creating Video Walls (page 31) for information on managing video walls in the built-in Maestro Z.

Syntax

```
create video-wall name
```

Parameters

direction

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 video-wall myWall
Success
```

Related Commands

```
delete video-wall
set video-wall
set video-wall-decoder
set video-wall-encoder
```

delete all-configuration

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

Syntax

```
delete all-configuration 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 all-configuration restart
```

Related Commands

`factory-defaults 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

`factory-defaults device`

delete multiview

Deletes the specified multiview from the database on the Management Platform. (ZyPer4K 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 multiview-window
set multiview
set multiview audio-source window number
show multiviews config
show multiviews status
```

delete multiview-window

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

Syntax

```
delete multiview-window 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 multiview-window myMultiview window 5
Success
```

Related Commands

```
create multiview
delete multiview
set multiview
set decoder multiview audio-source window number
show multiviews config
show multiviews status
```

delete video-wall

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

Syntax

```
delete video-wall name
```

Parameters

name

Type: **STRING**

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

Example

```
delete video-wall myWall
Success
```

Related Commands

```
create video-wall
set video-wall
set video-wall-decoder
set video-wall-encoder
```

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.

factory-defaults device

Set the specified device to the factory-default settings.

Syntax

```
factory-defaults device id
```

Parameters

id

Type: **STRING** or **MAC Address**

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

Example

```
factory-defaults device Airshow
Success

factory-defaults device 0:1e:c0:f6:a8:c3
Success
```

Related Commands

`delete all-configuration`

flash-leds

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 only)

Syntax

```
flash-leds id
```

Parameters

id

Type: **STRING** or **MAC Address**

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

Example

```
flash-leds myEncoder
Success

flash-leds 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.

- **analog-audio**
  Embeds audio stream from the encoder on the output of the decoder. The audio can be from the HDMI input or from the (analog) Audio jack on the encoder.
  In order to control what type of audio is being sent from an encoder, refer to the `set device` command.

- **fast-switched**
  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 only)

- **genlocked-scaled**
  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.

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

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

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

Syntax

```
join enc dec mode
```

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. String names are case-sensitive.

mode

Type: **STRING**

Supply one of the following arguments before executing this command.

argument	Description
`analog-audio`	Embed audio from the specified encoder
`fast-switched`	Join in “fast-switched” mode
`genlocked`	Low-latency join mode (ZyPer4K only)
`genlocked-scaled`	Low-latency with scale up/down (ZyPer4K only)
`hdmi-downmix-audio`	Join hdmi-downmix-audio to either hdmi-out or analog-out
`multiview`	Join in “fast-switched” mode (ZyPer4K only)
`video-wall`	Join in “fast-switched” mode
`window`	Join any portion of a source to any portion of a display (ZyPer4k only)
`usb`	Establish USB connection

Examples

```
join myEncoder1 myDecoder2 fast-switched
Success

join myEncoder1 myDecoder2 hdmi-downmix-audio hdmi-out
Success

join myEncoder1 myDecoder2 hdmi-downmix-audio analog-out
Success

join myMultiview2 Display4 multiview
Success

join myEncoder1 myWall video-wall
Success
```

Window Example

```
join myEncoder1 myDecoder2 window viewport-source 0 0 1920 1080
viewport-dest 500 500 500 500

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

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

join video-source

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

Syntax

```
join video-source 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)
hdmi-downmix-audio	automatically join hdmi-downmix-audio from connected encoder (ZyPer4K only)

Example

```
join video-source MyDecoder hdmi-downmix-audio
Success
```

Related Commands

channel

load encoder-edid

Uploads an EDID file to the specified encoder.

Syntax

```
load encoder-edid enc file only-lpcm-audio
```

Parameters

enc

Type: **STRING** or **MAC Address**

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

file

Type: **STRING**

The name of the file to load.

Example

```
load encoder-edid myEncoder myEDID.bin
Success
```

Related Commands

```
save device-edid
set server auto-edid-mode
```

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 down-servers

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

Syntax

```
redundancy delete down-servers
```

Parameters

none

Example

```
redundancy delete down-servers
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`

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
```

Related Commands

```
show server info
```

save device-edid

Saves the EDID of the downstream sink to the `srv/ftp/files` folder on the Management Server. Executing this command will generate two file types: `.bin` and `.txt`. The `.bin` file is the EDID is standard format. The `.txt` file is the decoded EDID data.
See Using Custom EDID Data (page 57) for more information on using this command.

Syntax

```
save device-edid 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 device-edid 0:1e:c0:f6:a5:2f myEDID
Success
```

Related Commands

`load encoder-edid`
`set server auto-edid-mode`

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 or RS232 string to the specified device. Use the *type* parameter to specify an IR or RS232 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 256 characters. (ZyPerUHD and ZyPer4K only)
cec	on | off (Used to turn a device on or off) (ZyPerUHD and ZyPer4K only)
cec hex-string	hex-numerals-no-delimiters (ZyPer4K only)
rs232	The string is ASCII and must not exceed 256 characters in length. Spaces and the followng control characters are supported as a portion of the string: \n New line \r Carriage return \t Tab \\ Slash \xnn Hex value, where nn is a two-digit hex value, including leading zeros

text

Type: **STRING**

The string to send. See the table, above, for restrictions.

Example

```
send myDecoder2 ir 000000690000001500 5f00170030001700300017003000 1700300017001700170030001700170017001700170030001700170017003000170 0300017001700170030001700170017001700170017001700300017003000170030 0200
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 hex-string command is not supported on ZyPerUHD

Related Commands

`set device rs232`

set encoder analog-audio-out

Sets the analog audio output source type for the specified encoder. (ZyPer4K and ZyPerUHD 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
`analog-audio-out`	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 hdmi-downmix`	Uses downmixed audio from input HDMI stream.

Example

```
set encoder Myencoder1 analog-audio-out source hdmi-downmix
Success
```

set encoder hdcp-mode

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
hdcp-mode	HDCP mode of the Encoder.

type

Type: **STRING**

The audio mode (analog or HDMI).

argument	Description
enabled	encoder will accept HDCP 1.4/2.2 compatible streams
disabled	encoder will reject HDCP 1.4/2.2 compatible streams.

Example

```
set encoder Myencoder1 hdcp-mode 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
analog-audio-out	Audio output from the Audio port on the decoder.
display-advanced-timing	Set advanced features, Front porch, sync width, sync polarity and total size
display-aspect-ratio	
display-mode	Set display to box, crop or stretch input stream within display resolution
display-size	Set display size manually (pixels) or automatically based on EDID.
hdcp-mode	Allows user to force HDCP protection at level 1.4 or 2.2 on previoulsy unprotected content.
hdmi-audio-out	Audio output from the HDMI port on the decoder.

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
force-version-1.4	Apply HDCP 1.4 protection to output stream
force-version-2.2	Apply HDCP 2.2 protection to output stream

The audio mode (analog out or HDMI out).

argument	Description
`source analog`	Uses the audio output created with the join command.
`source hdmi`	Uses the HDMI stream (HDMI audio-out only)
`source none`	No audio output
`source hdmi-downmix`	Uses the HDMI-downmix stream.

Display timing, aspect ratio, mode, size.

argument	Description
`sync-front-porch`	Synchronization mode.
`sync-width`	Synchronization width
`hsync-polarity`	Horizontal sync polarity (auto, negative, positive)
`vsync-polarity`	Vertical sync polarity (auto, negative, positive)
`total-size`	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)
`width`	Width in pixels or auto
`height`	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> display-size width <int> height <int> fps <int>

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 DF-BR1 display-size width 1280 height 720 fps 60

Command Description: Output display size determined by received EDID

Command Syntax
set decoder <Decoder_Name or MAC> display-size 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 < Decoder_Name or MAC > display-advanced-timing
sync-front-porch <pixelsHoriz:int>|auto <pixelsVert:int>|auto sync-width
<pixelsHoriz:int>|auto <pixelsVert:int>|auto
hsync-polarity positive|negative|auto
vsync-polarity positive|negative|auto
total-size <pixelsX:int>|auto <pixelsY:int>|auto

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 jagd-e1 display-advanced-timing
sync-front-porch 176 8
sync-width 88 10
hsync-polarity positive vsync-polarity positive
total-size 4400 2250

set decoder edid-prefer-mode

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
edid-prefer-mode	Select preferred EDID mode

type

Type: **STRING**

HDCP options. (Note: Valid with ZyPerUHD only) Used to minimize connection time.

argument	Description
max	Default mode. Selects the largest resolution defined in the EDID.
strict	Selects the Preferred resolution as stated in the display EDID

"**max**" – Default mode. Selects the largest resolution defined in the EDID. This has been the operating mode prior to this command. In almost all cases, this is the native resolution of the display. However, some displays can accept a resolution above the native (and scale down). In this case, it is better to use the "strict" mode.

"**strict**" – The Preferred Resolution is selected as defined in the EDID 1.3 specification. EDID 1.3 specifies that the first Detailed Timing Descriptor in the Standard Timing Information block is always the preferred resolution, although it is only the native resolution if the native-resolution flag is set. If the native-resolution flag is not set, then the maximum resolution will be chosen (falls back to "max" mode).

Note: All comparisons of "resolution" actually mean comparisons of the associated Pixel Clock. The Pixel Clock represents the entire resolution definition: horizontal and vertical size, fps, bit-depth and color decimation (RGB/4:4:4, 4:2:2, 4:2:0).

The command will immediately reanalyze the active EDID and if needed change the preferred resolution and reconnect to the encoder.

The reason for the "max" mode, and for it being the default, is that many displays do not follow the EDID 1.3 specification, claiming a native, Preferred Resolution below the display's actual native resolution. It is fairly common for a UHD display to have an HD resolution as the specified preferred resolution.

Note: ZyPer4K and ZyPerUHD, depending on mode, may support only a limited set of output resolutions, particularly when the scaler is enabled. ZMP will choose the active resolution based decoder capability, scaler mode and preferred resolution. However, the display's Preferred Resolution is displayed regardless of what the decoder ultimate actually uses. The active resolution is displayed in the decoder status as well.

Overriding Preferred Resolution Selection

It should rarely be required. But if the EDID supplied by the display is not correct, or for some reason ZMP chooses a Preferred Resolution that is not desired, the following command will force the decoder to a specific output resolution:

set decoder <decoder> display-size width <int> height <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-size command now has an option called "source".

set decoder <decoder> display-size 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 "edid-prefer-mode" 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 "display-size 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 "display-size" 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 genlock-scaled, video-wall or window mode, decoder FPS must equal encoder FPS
  - o Means 1080p60 scaling to UHD must be UHDp60, which won't work if display is only UHDp30 capable.
  - o If UHDp60 > decoder Preferred Resolution, then the output is left at 1080p60.
• If source is 1080i
  - o Output must be input FPS * 2
  - o If decoder resolution > 1080, it is set to 1080.

display-size = 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.

display-size = 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 display-size 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).

display-size = override
Same operation as display-size = 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.

display-size = 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-size = 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.

display-size = 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 display-size 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).

display-size = override

Same operation as display-size = auto, but effectively using a manually entered Preferred Resolution. Generally only used if the EDID is incorrect.

set decoder power-save

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 power-save 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 power-save enabled
Success
```

set device ethernet-management-port

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

Syntax

```
set device id ethernet-management-port 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 ethernet-management-port disabled
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device ip
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

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 analog-audio mode
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display 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
link-local	IP address self assigned Link-Local

Example

```
set device ip dhcp
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display 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 ip static 10.5.68.121 255.255.255.0 10.5.64.1
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

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		
9600	38400	115200
19200	57600	

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 analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set device send-ip-mcast-range

Sets allowable range of multicast addresses for selected devices. (ZyPer4K only)

Syntax

```
set device id send-ip-mcast-range 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 224.1.3.255

Example

```
set device encoders send-ip-mcast-range 224.1.1.25 224.1.2.125
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set device source-display iconImageName

Assigns an icon to the desired device. The icon will be displayed within the MaestroZ to identify the device.

Syntax

```
set device id source-display 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. Note that some filename extensions are `.JPG`.

argument	Description
`BluRayPlayer_Disc.png`	BluRay
`VinylRecord.png`	VCR
`HDDVD.png`	DVD
`VideoMixer.png`	Media Player
`CameraLens.png`	Camera
`SecurityCamera.png`	Security Camera
`WebCamera4.png`	Broadcast Camera
`digital-sinage.jpg`	Digital Signage Player
`PortableComputer.png`	Laptop
`Computer.png`	Desktop PC
`satellite-tv-1.png`	Cable TV Box
`2-satellite1.jpg`	Satellite Receiver
`3DDisplay.png`	Flat Panel Display
`video-projector.png`	Projector
`PlasmaDisplay1.png`	LED Wall
`led-ticker.png`	LED Ticker

Example

```
set device DVD-Wildlife source-display iconImageName HDDVD.png
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set device source-display location

Assigns a location description for the specified device.

Syntax

```
set device id source-display 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 source-display location VideoWall-1
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display iconImageName
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set device source-display manufacturer

Assigns a manufacturer description for the specified device.

Syntax

```
set device id source-display 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 source-display manufacturer Sony
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display model
set device source-display serialNumber
```

set device source-display model

Assigns a model description for the specified device.

Syntax

```
set device id source-display 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 numberof the device.
Do not use quotes when specifying this string value.

Example

```
set device myDecoder3 source-display model DVPSR210P
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display serialNumber
```

set device source-display serialNumber

Assigns the manufacturer serial number for the specified device.

Syntax

```
set device id source-display 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 source-display serialNumber 123456789
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device gig-ethernet-port mode
set device ip
set device ip static
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
```

set device usb-filter

Allows restrictions to USB use on selected device. (ZyPer4K only)

Syntax

```
set device id usb-filter 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
except-hid	Allows any USB device except HID devices
storage	Allows any USB device except Storage devices

Example

```
set device myDecoder2 usb-filter none
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device ip
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set device video-port

Selects active input port for ZyPer4K units with multiple inputs. (ZyPer4K only)

Syntax

```
set device id video-port 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
display-port	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 video-port display-port
Success
```

Related Commands

```
set device analog-audio mode
set device general name
set device ip
set device rs232
set device source-display iconImageName
set device source-display location
set device source-display manufacturer
set device source-display model
set device source-display serialNumber
```

set multiview

Assigns source to a position and size within a multiview display. (ZyPer4K only)

Syntax

```
set multiview id window-number wn encoder-name enc position-x posx
position-y posy size-x sx size-y 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 (1-9)

enc

Type: **STRING or MAC Address**

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

posx

Type: **Integer**

X coordinate of multiview window. Upper left corner of window. (0-99)

posy

Type: **Integer**

Y coordinate of multiview window. Upper left corner of window. (0-99)

sx

Type: **Integer**

Size/Length of multiview window. As a percentage of X dimension.

sy

Type: **Integer**

Size/Height of multiview window. As a percentage of Y dimension.

ly

Type: **Integer**

Window Layer. Value from 1-9 with layer 1 being the bottom layer and 9 being the top.

Example

```
set multiview myMview1 window-number 9 encoder-name myEnc1 position-x 50
position-y 50 size-x 25 size-y 25 layer 3
```

Related Commands

```
create multiview
delete video-wallmultiview
delete multiview-window
set device rs232multiview audio-source window-number
show multiviews config
show multiviews status
```

set multiview audio-source window number

Selects the input source to provide Audio for multiview display. (ZyPer4K only)

Syntax

```
set multiview id audio-source window number 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 1-9 identifying source to use for audio
none	Set no audio for the multiview window

Example

```
set multiview myMview1 audio-source window number 4
Success
```

Related Commands

```
create multiview
delete video-wallmultiview
delete multiview-window
set device rs232multiview
show multiviews config
show multiviews status
```

set responses rs232-term-chars

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 specifed, 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 decoder-2 rs-232-term-chars "\r"
Success
```

Related Commands

`set device rs232`

set server api-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. Note that Maestro Z will always display both the Username and Password fields. The username and password for Maestro Z are both `admin`. As of this writing, the username or password for Maestro Z can *not* be changed. This functionality will be added with a future software update.

Syntax

```
set server api-password pass
```

Parameters

pass

Type: **STRING**

The desired password.

Example

```
set server api-password biGB055
Success
```

Related Commands

`set server auto-edid-mode`
`set server timezone`

set server auto-edid-mode

Sets the EDID mode for the Management Platform. By default, Auto-EDID mode is enabled.

Syntax

```
set server auto-edid-mode 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 auto-edid-mode disabled
Success
```

Related Commands

```
set server api-password
set server timezone
```

set server enhanced-logging

Sets the server to do additional logging of events. By default, enhanced-logging is disabled. This feature should only be enabled at the direction of ZeeVee Technical Support. ZyPer4K only.

Syntax

```
set server enhanced-logging mode reboot
```

Parameters

mode

Type: **STRING**

Supply one of the following arguments.

argument	Description
disabled	Disables enhanced logging mode.
enabled	Enables enhanced logging mode.

Example

```
set server enhanced-logging enabled reboot
Success
```

Related Commands

```
set server api-password
set server timezone
```

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 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 reboot
Success
```

Related Commands

```
set server api-password
set server timezone
```

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 api-password
set server auto-edid-mode
```

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 virtual-ip address IP_Address
network-interface video|management
```

Parameters

serv_id

Type: **STRING**

The servers to apply Virtual-ID to.

argument	Description
`all-servers`	All Management Platforms on the Network. (Master and Slave)
`this-server`	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
`network-interface`	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 all-servers virtual-ip address 192.168.0.25
network-interface v
Success

set server redundancy this-server preferred-master true preferred-
slave false
Success

set server redundancy 192.168.1.202 preferred-master false
preferred-slave true
Success
```

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
```

Related Commands

```
set server api-password
set server auto-edid-mode
```

set video-wall

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 video-wall` *`id rows cols bezt bezb bezl bezr`*

Parameters

param1

Type: **STRING**

The name of the video wall. String names are case-sensitive.

rows

Type: **INTEGER**

The number of rows. (Maximum 5 for ZyPer4K/ZyPerUHD, Maximum 4 for ZyPerHD)

cols

Type: **INTEGER**

The number of columns. (Max 5 for ZyPer4K/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

Example

```
set video-wall myVideoWall 4 4 10 10 15 15
Success
```

Related Commands

```
create video-wall
set video-wall
show video-walls
join video-wall
set video-wall new-name
```

set video-wall decoder

Assigns the specified decoder, to the desired row and column, on the specified video wall.

Syntax

```
set video-wall 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 video-wall myVideoWall decoder myDecoder 2 3
Success
```

Related Commands

`create video-wall`
`set video-wall`
`show video-walls`
`join video-wall`

set video-wall new-name

Changes the name of an existing video wall

Syntax

```
set video-wall id new-name 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 video-wall myWall2 new-name yourWall2
Success
```

Related Commands

```
create video-wall
set video-wall
show video-walls
join video-wall
```

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(d8:80:39:9a:e6:d);
  device.CapabilitiesVersion; values=1
  device.streams-supported; values=video:video-scaled:analog-
audio:hdmi-downmix-audio
  device.stream-mode-settable; values=video:video-scaled:analog-
audio:hdmi-downmix-audio
  device.stream-mcast-settable; values=video:analog-audio:hdmi-
downmix-audio
  device.join-video; values=multiview:genlocked:fast-switched
  device.video-port; values=hdmi
  device.hdmi-status; values=link:hdcp:resolution:fps
  device.join-audio; values=analog:hdmi-downmix-audio
  device.hdmi-audio-source; values=analog:hdmi:hdmi-downmix
  device.analog-audio-source; values=none:analog:hdmi-downmix
  device.encoder-analog-audio-source; values=none:hdmi-downmix
  device.ir; values=device:server
  device.rs232; values=device:server
  device.video-wall; values=maxSize(5):bezelsSupported
  device.send-multicasts; values=settable
  device.ip-mode; values=dhcp:static
  device.ip-params; values=address:mask:gateway
  device.decoder-edid; values=save
  device.encoder-edid; values=save:load
  device.flash-leds; values=supported
  device.ethernet-management-port-mode; values=supported
  device.video-input-port; values=hdmi:display-port:auto
lastChangeIdMax(5072370);
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.

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 config USA4K since 20
device(d8:80:39:9a:e6:d);
  device.gen; model=Zyper4K, type=encoder, name=USA4K, state=Up,
lastChangeId=12627
  device.gen; firmware=3.1.3
  device.gen; ethernet-management-port-mode=enabled
  device.ports; video-port=hdmi
  device.ip; mode=dhcp, address=172.16.6.33, mask=255.255.255.0,
gateway=172.16.6.1
  device.rs232; sendingToMacOrIp=none(0.0.0.0), terminationChars=\
n\r,  baudrate=57600, dataBit=8, stop_Bit=1, parity=none
  device.ir; sendingToMacOrIp=none(0.0.0.0)
  device.source; iconImageName=GenericVideoSource
, manufacturer=Unknown, model=Unknown, location=Unknown,
serialNumber=Unknown
  device.audioOutSourceType; analogOutSourceType=none
  device.sendIpMcastRange; first=224.1.1.1, last=224.1.3.255
  device.videoStream; ipMcastAddr=224.1.1.7, mode=disabled,
datarate=0Mbps
  device.videoScaledStream; ipMcastAddr=224.1.1.4, mode=enabled,
datarate=615Mbps
  device.analogAudioIpMcast; ipAddr=224.1.1.9, mode=enabled
  device.audioDownmixIpMcast; ipAddr=224.1.1.8, mode=disabled
lastChangeIdMax(132613);
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 USA4K since 20
device(d8:80:39:9a:e6:d);
  device.gen; model=Zyper4K, type=encoder, name=USA4K, state=Up,
uptime=1d:4h:36m:37s, lastChangeId=5285
  device.firmwareUpdate; status=idle, loadingFile=none,
percentComplete=0
  device.hdmiInput; cableConnected=connected, hdcp=inactive, hdcp-
version=NONE, hdmi-2.0=yes, horizontalSize=3840, verticalSize=2160,
fps=30.00, interlaced=no
  device.hdmiData; color=YUV_4:4:4, bits-per-pixel=8,
datarateMbps=615
  device.edid; status=valid
  device.autoEdid; decoder-name=not-enabled
lastChangeIdMax(109978);
Success
```

Related Commands

`show device config`

show multiviews config

Shows configuration information on all multiview displays. (ZyPer4K 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 video-wallmultiview
delete multiview-window
set device rs232multiview
show multiviews status
```

show multiviews status

Shows status information for all multiview displays. (ZyPer4K only)

Syntax

```
show multiviews status
```

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 video-wallmultiview
delete multiview-window
set device rs232multiview
show multiviews config
```

show responses

Displays response strings from the specified device.

Syntax

```
show responses id type param3
```

Parameters

id

Type: **STRING or MAC Address**

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

type

Type: **STRING**

Supply one of the following arguments.

argument	Description
`ir`	Displays IR response strings.
`rs232`	Displays RS232 response strings.

param3

Supply one of the following arguments.

argument	Description
`last`	Displays the last received response, based on the argument supplied for the *`type`* parameter.
`last-change-id`	Displays the **`lastChangeId`** of the most recently received response.
`since`	Displays only new response data. Follow this argument with desired value to query.

Example

```
show responses 0:1e:c0:f6:b0:8a rs232 since 10
lastChangeId(0);
Success

show responses 0:1e:c0:f6:b0:8a ir last-change-id
lastChangeId(0);
Success

show responses 0:1e:c0:f6:b0:8a ir last
lastChangeId(0);
Success

show responses Dec rs232 last
device(34:1b:22:80:57:f8);
  device.rs232Response.7; string=”Yes ZeeVee Support is the
Greatest\r\n”
lastChangeId(7);
Success
```

show server config

Displays configuration information for the Management Platform.

Syntax

```
show server config
```

Parameters

none

Example

```
show server config
server(172.16.6.46);
  server.gen; autoEdidMode=disabled
  server.multicast; startLSB:1
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(192.168.4.202);
  server.gen; hostname=zyper.local, version=1.8.34236,
previousVersion=1.8.34169, macAddress=40:8d:5c:32:44:f3,
serialNumber=ZZM1G600011B
  server.gen; uptime=0d:7h:17m:17s, freeMem=6.933GB, bootCount=318
  server.gen; runningInVm=false
  server.time; time=Thu Jul 12 15:04:54 2018, timezone=America/
New_York
  server.license; productID=038D0240-045C-0532-4406-F30700080009,
license=RNVC-LHDL-0050-2F34-8709-FF47-AE75-AD3E-CF6E-2CA2-5585
  server.license; limit=50, knownDevices=26, devicesUp=23,
devicesExceeded=0
  server.deviceUpdates; active=0
  server.activeDeviceVersions; num_0.5.5=2, num_0.9.4=3,
num_2.11.4=2, num_2.12.2=8, num_2.12.3=1, num_3.5.2=7
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(192.168.1.201);
  server.status; state=slave, version=1.7.1.33348, wasMaster=true,
wasSlave=true
  server.config; preferredMaster=true, preferredSlave=false
  server.virtualIp; address=192.168.1.206, mask=255.255.255.0
  server.slaveStatus; dbRunning=Yes, dbRunningState=Slave has read
all relay log; waiting for the slave I/O thread to update it, G
TID=0-1546798323-2301091, dbError=none
server(192.168.1.202);
  server.status; state=master, version=1.7.1.33348, wasMaster=true,
wasSlave=true
  server.config; preferredMaster=true, preferredSlave=false
  server.virtualIp; address=192.168.1.206, mask=255.255.255.0
Success
```

Related Commands

```
set server redundancy
redundancy switchover
```

show video-walls

Displays all video walls that have been created and all associated information.

Syntax

```
show video-walls
```

Parameters

none

Example

```
show video-walls
videoWall(myWall);
  videoWall.gen; videoSourceMac=0:1e:c0:f6:a8:c3, numDisplayRows=4,
numDisplayCols=4
  videoWall.bezel; top=0, bottom=0, left=0, right=0
  videoWall.decodersRow1; col1=none, col2=none, col3=none,
col4=none;
  videoWall.decodersRow2; col1=none, col2=none, col3=none,
col4=none;
  videoWall.decodersRow3; col1=none, col2=none, col3=none,
col4=none;
  videoWall.decodersRow4; col1=none, col2=none, col3=none,
col4=none;
Success
```

Related Commands

```
create video-wall
set video-wall
set video-wall-decoder
set video-wall-encoder
```

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 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
analog-audio	analog audio multicast stream.
hdmi-downmix-audio	downmix audio multicast stream
video	full scale video stream
video-scaled	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 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
analog-audio	analog audio multicast stream.
hdmi-downmix-audio	downmix audio multicast stream
video	full scale video stream
video-scaled	downscaled video stream (for multiview)

Example

```
stop encoder Myencoder1 stream video-scaled
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
`ir`	Specifies a IR connection. (ZyPer4K only)
`rs232`	Connection to another device or the server. Set *`rxid`* = `none` to pass data to an arbitrary IP host.

Example

```
switch DVD-Wildlife SonyXBR4 rs232
Success
```

Related Commands

`send`

trouble-report

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

```
trouble-report
```

Parameters

none

Example

```
trouble-report
Success
```

update server

Updates the Management Platform software. The server software file uses the `.zyper` extension. Refer to Updating the Software (page 167) for more information on using this command.

Syntax

```
update server file
```

Parameters

file

Type: **STRING**

The full filename of the software file.

Example

```
update server new-sofware-file.zyper
Success

Server rebooting; connection will end
```

4 Event Mechanism

events

ZMP Event Mechanism

There are three ways to receive events:
• Second telnet session to receive events asynchronously. Session not used for API commands, only to receive events.
• Browser WebSocket to ZMP server. Allows server to asynchronously send events to the browser.
• Reliable, low-overhead API command to poll for events.

Event Message Format

```
Event::<name>::<source>::<date>::<lastChangeId>::: <Message>
```

Where:

source device-name or “server”

Example:

Event::DeviceStateChange::DE1(d8:80:39:9a:af:e1)::Jun-18-02:42:56:PM::13:::
state=Up

Telnet Event Session

• Client telnets to the ZMP server as normal
• API prompt received
• Command entered: “events”
  o Causes the event mode to be entered
  o Server sends initial events (described below) and new events as they occur to this telnet session
  o Any character entered to the server causes the mode to exit back to the API prompt

Browser WebSocket

Client usage of a WebSocket to receive events is quite simple. Example JavaScript from the sample zyper.html file shows how to connect to the websocket server on ZMP. Upon connection, initial events (described below) will be sent, and then any new events as they occur.

```
eventSock = new WebSocket(“ws://rey:8001”, “zeeVeeLogging”
eventSock.onopen = eventSockOpened;
eventSock.onMessage = eventRcvd;
eventSock.onclose = eventSockClosed;

function eventSockOpened() {
        eventSock.send(“Send Events”);    // igonored by server
}
function eventRcvd(event) {
        var evWin = document.getElementById(‘eventWindow’);
        evWin.innerHTML += event.data + “<br>”;
        evWin.scrollTop = evWin.scrollHeight;
}
function eventSockClosed() {
        console.log(“EVENT SOCK CLOSED”);
}
```

API Polling

The “show events since <id>” command may be used over telnet or from a browser using AJAX/JSON and preferably long-polling. It is a simple, low overhead and very reliable mechanism to ensure all events have been received.

```
Zyper$ show events since 0
server(172.16.2.169);
   server.event.0; event=”Mon Jun 18 18:44:06 2018: ip=172.16.2.64, state=down-
   >master”
   server.event.1; event=”Mon Jun 18 18:44:10 2018: state=up”
   server.event.2; event=”Mon Jun 18 18:44:10 2018: state=up”
   ...
   server.event.28; event=”Tue Jun 19 05:00:03 2018: sizeX=1280->720,
   sizeY=720->480, fps=60.00->60.00”
   server.event.29; event=”Tue Jun 19 05:01:24 2018: cable=disconnected”
   server.event.30; event=”Tue Jun 19 05:01:26 2018: cable=connected”
   server.event.31; event=”Tue Jun 19 05:01:26 2018: sizeX=720->1280,
   sizeY=480->720, fps=60.00->60.00”
lastChangeId(32);
Success
Zyper$
Zyper$ show events since 28
server(172.16.2.169);
   server.event.28; event=”Tue Jun 19 05:00:03 2018: sizeX=1280->720,
   sizeY=720->480, fps=60.00->60.00”
   server.event.29; event=”Tue Jun 19 05:01:24 2018: cable=disconnected”
   server.event.30; event=”Tue Jun 19 05:01:26 2018: cable=connected”
   server.event.31; event=”Tue Jun 19 05:01:26 2018: sizeX=720->1280,
   sizeY=480->720, fps=60.00->60.00”
lastChangeId(32);
Success
Zyper$
Zyper$ show events since 32
lastChangeId(32);
Success
Zyper$
```

Initial Events

Upon entering telnet “events” mode, or upon a WebSocket connection, the server will send a DeviceStatus event for each known device. Each of these events will have lastChangeId set to 0.

Event List

ServerIpChanged:

Message: `ipAddress=<from>-><to>`

Example:

Event::ServerIpChanged::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
ipAddress=169.254.1.10->172.16.2.22

ServerStateChange:

Message: `state=<from>-><to>`

serverState:

down
initialization
master
slave-sync
slave
slave-switching-over
slave-db-updating
not-participating
slave-waiting-for-master
slave-version-mismatch

Example:

Event::stateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=slave->master

NewServer:

Message: `id=<id>, ip=<ipAddr>, state=<serverState>`

Example:

Event::NewServer::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: id=
693EF360-B908-11DC-9EA5-10BF483EF417, ip=172.16.2.22, state=initialization

OtherServerStateChange:

Message: `ip=<ipAddr>, state=<from>-><to>`

Example:

Event::OtherServerStateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-
02:42:56:PM::15::: ipAddress=169.254.1.10->172.16.2.22, state=slave->master

DeviceStatus: initial device state

Message: `state=down`
Message: `state=up, uptime=<seconds>, cable=disconnected`
Message: `state=up, uptime=<seconds>, cable=connected, sizeX=<pixels>, sizeY=<pixels>, fps=<fps>`
Message (DECODER): `state=up, uptime=<seconds>, cable=connected, sizeX=<pixels>, sizeY=<pixels>, fps=<fps>, receivingVideoFromEncoder=no|yes|yes-with-warning [, reason=<reason> | warning=<warning>]`

“no” reason list:

decoder not joined
decoder down
decoder hdmi down
encoder down
encoder stream disabled
encoder hdmi down
display does not support resolution
encoder and decoder hdcp do not match
encoder has unsupported color format
encoder data rate exceeded
decoder data rate exceeded
multiview error -- do ‘show multiviews status’
decoder resolution < UHD; can’t join multiview
encoder resolution < UHD; can’t join video-wall
encoder has multiview conflict with genlock
problem with network connection

“yes-with-warning” warning list:

multiview partially active -- do ‘show multiviews status’

Examples:

Event::DeviceState::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=down
Event::DeviceState::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=up, uptime=1234, cable=connected, sizeX=3840, sizeY=2160, fps=60,
receivingVideoFromEncoder=no, reason=encoder hdmi down

DeviceStateChange: device up/down

Message: `state=up|down`
Example:

Event::DeviceStateChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
state=up

CableConnection

Message: `cable=connected|disconnected`
Example:

Event::CableConnection::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15:::
cable=connected

VideoStatusChange: sending video or not and why

Message: `receivingVideoFromEncoder=no|yes|yes-with-warning [, reason=<reason> | warning=<warning>]`

Example:

Event::VideoStatusChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: receivingVideoFromEncoder=no, reason=encoder hdmi down

ResolutionChange: just resolution change

Message: `sizeX=from->to, sizeY=from->to, fps=from->to`

Example:

Event::ResolutionChange::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: sizeX=1920->3840, sizeY=1080->2160, fps=59.95->59.94

VideoChanged: encoder only, non-resolution change

Message: `interlaced=<from>-><to>, color=<from>-><to>, colorDepth=<from>-><to>, hdcp=<from>-><to>, hdcpVersion=<from>-><to>, hdmi20=<from>-><to>`

Where:

Interlaced: yes, no
Color values: RGB, YUV444, YUV422, YUV420
colorDepth: 8, 10, 12
hdcp: yes, no
hdcpVersion: none, 1.4, 2.2
hdmi20: yes, no

Example:

Event::VideoChanged::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: interlaced=yes->no, color=RGB->YUV444, colorDepth=8->10, hdcp=yes->no, hdcpVersion=1.4->2.2, hdmi20=no->yes

RS232Data

Message: `data=<rs232Data>`

Example:

Event::RS232Data::EE5(d8:80:39:9b:c:e5)::Jun-18-02:42:56:PM::15::: data=”hello there”

Note: RS232 events are only sent after a termination character has been received. If there are no termination characters defined, an event is generated after 10ms of no additional input.

IRData

Message: `data=<irData>`

Example:

Event::IrData::DCD(d8:80:39:9a:d0:cd)::Jun-19-05:02:07:PM::133:::
data=\"0000006d0000002700ae00a70016000f00150010001500f00150
0100015000f00150034001600 0f0015000f0015000f00150010001400100
01400100015000f00140010008001600350016003400160034001600 0f0
01500100015000f00150010001500 0f00150010001400100014001000 1
5000f00150034001600330016003400160034001600330016003300160-
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

5 Appendix

Updating the Software

Using Mac OS X

1. Make sure the Management Platform is powered and is working correctly.

2. Download the latest software from the ZeeVee website. Make note of the location of where the software was downloaded.

3. Launch the Terminal app, found under the Applications > Utilities folder. By default, the current directory will be the Home directory.

```
Last login: Tue Mar 22 14:24:08 on console
Andrews-MacBook-Pro:~ Andrew$
```

4. Change the directory to the location of the downloaded software file. For example, if the software was downloaded to the Desktop, then change to the Desktop folder, as shown:

```
Last login: Tue Mar 22 14:24:08 on console
Andrews-MacBook-Pro:~ Andrew$ cd desktop
Andrews-MacBook-Pro:desktop Andrew$
```

5. Use the FTP protocol to login to the Management Platform. At the terminal prompt, type the following and press the [ENTER] key.

```
Andrews-MacBook-Pro:desktop Andrew$ ftp 192.168.1.6
```

6. Enter the user name and password. Use `anonymous` for the user name and use `guest` for the password. The password will not be echoed to the screen.

```
Andrews-MacBook-Pro:desktop Andrew$ ftp 192.168.1.6
Connected to 192.168.1.6
220 (vsFTPd 3.0.2)
Name (192.168.1.6:Andrew): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

7. Type `cd files` at the ftp prompt to change to the `/files` directory.

```
ftp> cd files
250 Directory successfully changed.
ftp>
```

8. Enter and run the `put` command, followed by the full name of the software file, as shown. Make sure to replace [version] with the version of the filename you are using. For example:

```
ftp> put update_nuc_1.7.1.33348.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_1.7.1.33348.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
   ```

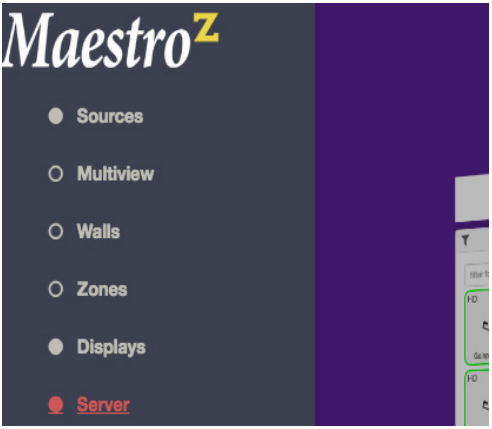
Using MaestroZ

1. Make sure the Management Platform is powered and is working correctly.

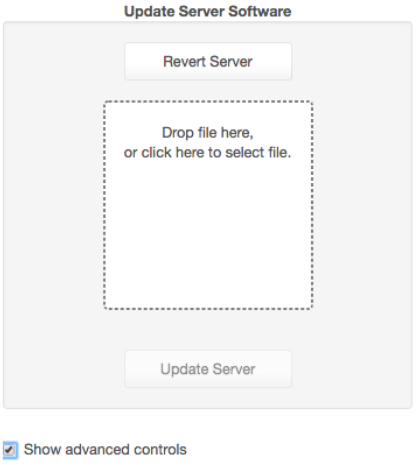
2. Download the latest software from the ZeeVee website. Make note of the location of where the software was downloaded.

3. Login to the Maestro Z. Refer to Accessing Maestro Z (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.



Redundancy Configuration Instructions

To configure redundancy, follow the steps below. The secondary server must be running for the redundancy fields to be visible in MaestroZ 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 MaestroZ

1) Login through you Master ZMP MaestroZ with Chrome.

2) Open the Server Panel

3) Scroll down to the Redundancy fields

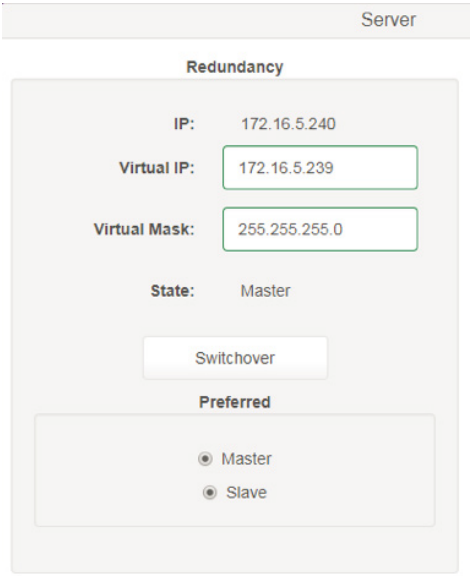
4) Set the fields listed below.

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 MaestroZ.

Virtual Mask: The subnet mask for the virtual interface, must be correct for the IP address listed above and not it should not conflict with the main eth0 interface.

Preferred Roles Radio Button: The preferred roles for the server. This field is used to decide the Master or Slave upon both servers initializing at the same time. Although rare, this can occur.

State: The current role of the current Server connected to.



After configuration is complete on the Master, the information should populate to the Slave server. The preferred roles for the Slave server will still need to be configured. This can be done by logging into MaestroZ 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 MaestroZ 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	• Intel® Celeron® Processor J4005
Operating System	• Linux Ubuntu 14.04
Internal Storage	• 60 GB SSD
Graphics	• Intel® HD Graphics 600
LAN	• Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• Input: 100 ~ 240 V AC • Output: 19V DC, 3.42 A
I/O	• 2 x HDMI 2.0a • 4 x USB 3.0, Type- A, female • 1 x RJ45 • 1 x 19V DC • 1 x Kensington lock slot • 2 x 3.5mm headset jacks (Not used)
Operating Temperature	• 0 ºC to +40 ºC
Storage Temperature	• -20 ºC to +60 ºC
VESA	• VESA Bracket included Supports 75 x 75 and 100 x 100 mm
Dimensions (W x H x D)	• 4.55 in x 2.01 in x 4.57 in (115 mm x 51 mm x 111 mm)

Hardware Specifications (Enterprise Grade Rack Mount)

CPU	• Intel® Xeon E3-1200 v5
Operating System	• Linux Ubuntu 14.04
Internal Storage	• 64 GB SSD
Graphics	• ASPEED AST2400 BMC
LAN	• Dual Gigabit LAN
Internal Memory	• 8 GB DDR4
Power Supply	• 200W Low-Noise AC-DC power supply with PFC
I/O	• 1 x VGA (15-pin D-sub) • 2 x USB 2.0, Type-A, female • 2 x RJ45 (LAN) (Video Network and Management Network) • 1 x RS232 (9-pin D-sub)
Operating Temperature	• +10 ºC to +35 ºC
Storage Temperature	• -40 ºC to +70 ºC
Dimensions (W x H x D)	• 17.2 in x 1.7 in x 11.3 in (437 mm x 43 mm x 287 mm)
Weight	• 8.45 lbs, (3.83 kg)

c UL us CE RoHS FC

Ethernet Port 0 = Video Port. DCHP default IP Address
Etherent Port 1 = Managment Port.
Static IP Address 192.168.20.2 Subnet Mask = 255.255.255.0

zeevee
Signal Intelligence

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

ZyPer_ManagementPlatform_Manual_RevB 2016-05