# NETWORK HD

## AV over IP Encoders (TX) and Decoders (RX)
### 100 | 200 | 400 | 500 | 600 Series

Application Programming Interface

| Document Revision: | v6.5 |
|---|---|
| Document Date: | January 2023 |

# 1.  Overview

The following document contains the Application Programming Interface (API) commands to communicate with and control WyreStorm NetworkHD systems via serial and IP commands. Read this document in its entirety before starting any communication with the NetworkHD products.

## 1.1  Document Use of Terms

| Term | Description |
|---|---|
| NHD-CTL | NHD-000-CTL or NHD-CTL-PRO Controller for NetworkHD Systems |
| Encoder | NetworkHD device that accepts Video, Audio, Infrared, RS-232 and USB signals and generates AV over IP media streams |
| Decoder | NetworkHD device that accepts AV over IP media streams and generates Video, Audio, Infrared, RS-232 and USB signals |
| Endpoint | A NetworkHD encoder or decoder |
| TX | A NetworkHD encoder |
| RX | A NetworkHD decoder |
| Hostname | A fixed name stored in an endpoint's firmware that is hard coded based on the model and the MAC Address<br><br>Example: NHD-100-TX-AABBDDCCEEFF |
| Alias | Custom name given to a NetworkHD endpoint for easy management. These names are assigned using the NetworkHD Console Software |
| IR | Infrared |
| Video Wall | An array of video displays used together to form a larger logical video wall group treated as a single reference |
| Scene | A particular configuration of a video wall, including assignment of encoders and decoders, configured and saved using the Console utility |
| Standard Video Wall | A type of video wall Scene that includes decoders with a single source assignment and Logical Screens |
| Video Wall Withing a Wall | A type of video wall Scene that uses the concept of windows and layers |
| Logical Screen | An element of a 'standard video wall' Scene where 2 or more physical displays are joined to show a single source across them |
| Window | An element of a 'video wall within a wall' Scene where 1 or more physical displays are joined to show a single source across them |
| Multiview | A concept where multiple video sources are composited on single display in separate tiles using Layouts. An element of a Multiview layout that shows an area of the canvas where a single video source is displayed |
| Canvas | An element of a Multiview layout. The Canvas is the total area available in which to place Tiles |
| Tile | An element of a Multiview layout that shows an area of the canvas where a single video source is displayed |
| Layer | The layer order defines which source windows (video walls) or tiles (Multiview) are in front of or behind other windows/tiles. This is an important consideration where windows/tiles overlap |

## 1.3 NetworkHD Communication Topology

Most communications from a 3rd-party control system are exclusively interfaced to NetworkHD products using the NHD-CTL Controller for NetworkHD Systems. The NHD-CTL is a communications bridge for Telnet or RS-232 data ingress and egress from a NetworkHD system.

NetworkHD connectivity to 3rd-party control systems is established via a single IP address or RS-232 port, a connection established with the NHD-CTL. For general control API communications, a 3rd-party control system will have no direct communications with a single NetworkHD TX or RX endpoint.

One exception to this single point of communication concerns NetworkHD TX source video preview streams. Preview streams use a HTTP structure for API communications and encapsulation of the video preview stream. Video preview streams are not processed by the NHD-CTL Controller and communications are sent directly to a 3rd-party processor or UI from a NetworkHD encoder. Source video preview streams are detailed in section 14 – Video Preview Streams.

The diagram below shows a NetworkHD API communication topology example:
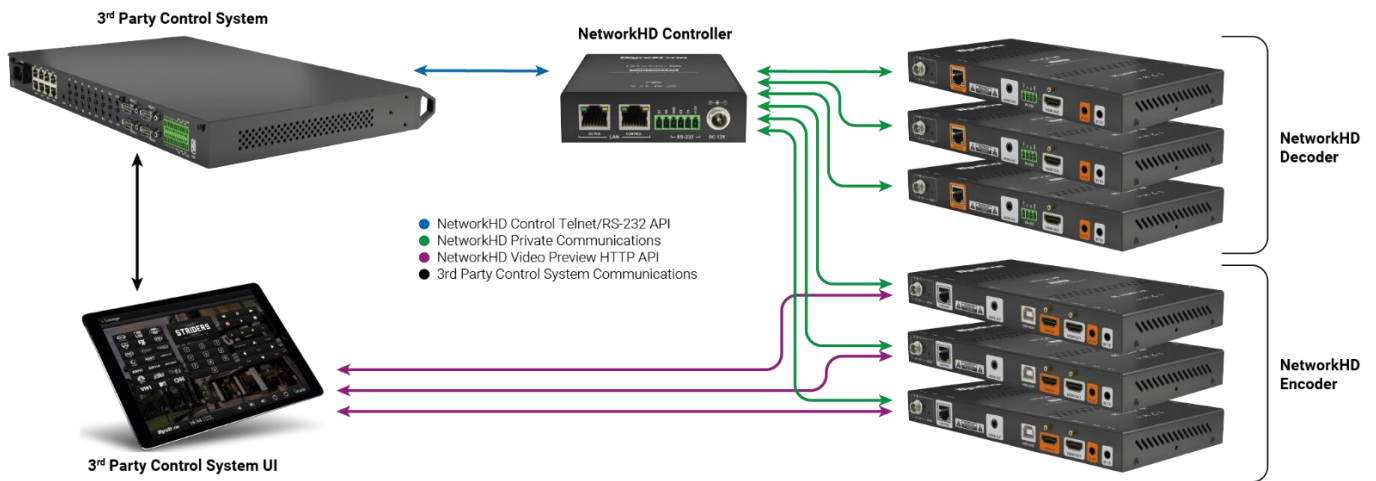
# 2.     Communication Wiring and Configuration

WyreStorm recommends that all wiring for the installation is run and terminated prior to making connections to NetworkHD products. Read through this section in its entirety before running or terminating any cables to ensure proper operation and to avoid damaging equipment.

Communication and control of a NetworkHD system is established by connecting to the NHD-CTL controller. A 3rd-party control system will connect to the NHD-CTL Controller.

**NHD-000-CTL**



| Option 1 | Option 2 | Option 3 |
| Network API | Network API | RS-232 API |

**NHD-CTL-PRO**



**Option 1**
Network API

**Option 2**
Network API

**Option 3**
RS-232 API
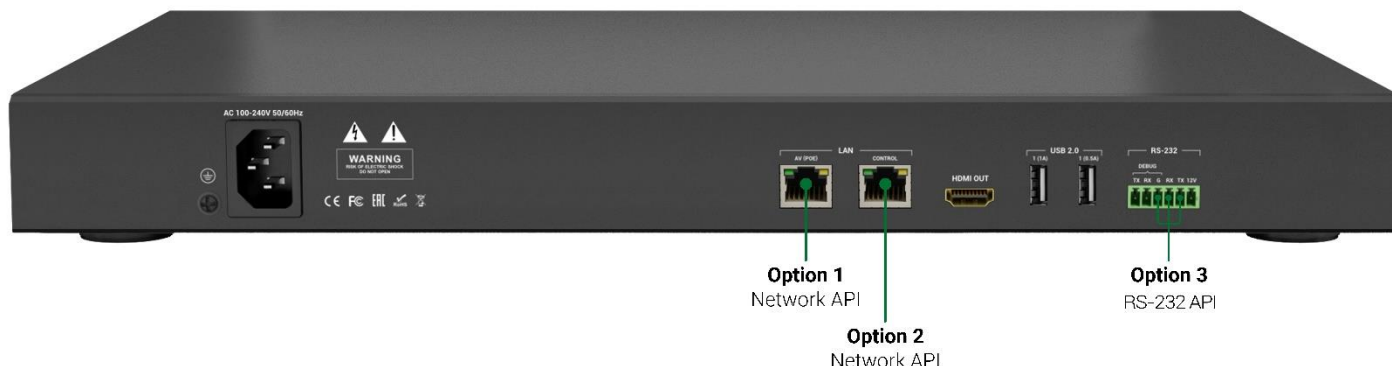
## 2.1  IPv4 Network Communications

The NHD-CTL has two IP network interfaces. API communications are achieved by connecting with either one of the network interface ports on the NHD-CTL.

When using factory default settings, the two NHD-CTL network interfaces have the following parameters:

| Port Label | IPv4 Address | Subnet Mask | Default Gateway IPv4 Address |
| --- | --- | --- | --- |
| AV | 169.254.1.1 | 255.255.0.0 | 169.254.1.1 |
| CONTROL | 192.168.11.243 | 255.255.255.0 | 192.168.11.1 |

## 2.2 LAN Protocols

By default, the NHD-CTL uses Telnet (RFC 854) protocol.

Additional communication protocols can be enabled manually via the CTL's web interface.

| | Telnet (Default) | Telnet over TLS (SSL) | SSH |
| --- | --- | --- | --- |
| **Port** | 23 | 992 | 10022 |
| **Credentials** | <none> | User: wyrestorm<br>Pass: networkhd | User: wyrestorm<br>Pass: networkhd |

## 2.3 RS-232 Serial Communications

The NHD-CTL has a single RS-232 port for 3rd-party control. API communications are achieved by connecting a 3rd-party RS- 232 communications port directly to the RS-232 port on the NHD-CTL.

The wire connections to 3rd-party RS-232 equipment is as follows:

| NHD-CTL RS-232 Pin | Function | Cable | 3rd-party RS-232 Pin | DE-9M Pin | DE-9F Pin | Function |
|---|---|---|---|---|---|---|
| TX | Transmit | ✗🖉➜ | RXD | 2 | 3 | Receive |
| RX | Receive | ✗🖉➜ | TXD | 3 | 2 | Transmit |
| GND | Signal Ground | ✗🖉➜ | GND | 5 | 5 | Signal Ground |

Where 3rd-party control equipment has a 9-pin D-subminiature connector, be sure to correctly identify if this connector is male (DE- 9M) or female (DE-9F) as the pin functions are different.
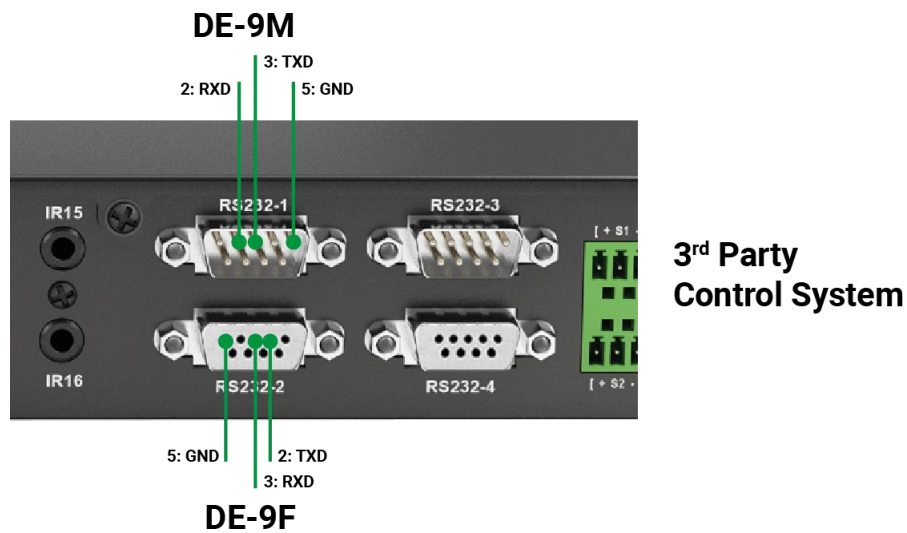


The RS-232 port on the NHD-CTL uses the following parameters:

| Type | Baud Rate | Data Bits | Parity Bits | Stop Bits | Flow Control |
|---|---|---|---|---|---|
| 3 Wire RS-232 | 9600 | 8 | None | 1 | None |

# 3.    Command Overview

## 3.1    Command Delimiter for Sent Commands

When sending commands using the IPv4 / Telnet API channel, or when using the RS-232 API channel, all command lines sent from the 3rd-party controller to the NHD-CTL should end with a specific character. This signifies when the command is processed by the NHD-CTL. This is usually specified in 3rd-party control software as the "command delimiter," "stop character," or "line terminator."

Accepted delimiter characters are:

| Character | Shorthand | Hex Notation | Escape Notation | Decimal Notation |
|---|---|---|---|---|
| Line Feed | LF | 0A | \n | 10 |
| Carriage Return + Line Feed | CR LF | 0D 0A | \r\n | 13 10 |

Please note, most 3rd-party control software will either append these characters automatically or an option to specify them will be present.
It is important that the last delimiter character is LF and not CR.

## 3.2    Command Delimiter for Received Data

When receiving command responses or notifications using the IPv4 / Telnet API channel, or when using the RS-232 API channel, all data sent from the NHD-CTL to the 3rd-party controller will end with a specific character order. This signifies when the return or notification data from the NHD-CTL is complete.

The delimiter characters are:

| Character | Shorthand | Hex Notation | Escape Notation | Decimal Notation |
|---|---|---|---|---|
| Carriage Return + Line Feed | CR LF | 0D 0A | \r\n | 13 10 |

## 3.3    Command Response

Upon receiving a valid command from the 3rd-party control system, the NHD-CTL may send confirmation data or return data in response. The type of response and the data sent from the NHD-CTL will depend on the preceding command sent to it. Please see the specific API command for details.

## 3.4    Use of Quotation Marks / Inverted Commas

Within an API command where the character " is shown, this will explicitly reference the standard ASCII symbol using hexadecimal byte value 0x22. No other symbol is to be used, including other inverted comma symbols or quotation mark symbols. If other inverted comma or quotation mark symbols are used, the NHD-CTL will treat the command as incorrect data (See section 3.5).

The correct inverted comma symbol can be copied from here:

## 3.5    Unknown Command or Incorrect Data

If the 3rd-party controller sends an unknown command, or command parameters that are incorrect or unreferenced, the NHD-000- CTL will send the following response:

```
unknown command
```

## 3.6    API Notifications

The server may react to a change in system status or matrix assignment by sending data to the 3rd-party controller asynchronously, that is without the 3rd-party controller first sending a command to the NHD-CTL. API notifications can occur at any time.

Please see section 12 - API Notifications for details.

## 3.7    Device Hostname and Alias

The NetworkHD API can reference a NetworkHD endpoint (encoder or decoder) by using either the device Hostname or the device Alias.

The device Hostname is hard coded into the NetworkHD endpoint and will never change. If the endpoint is swapped out for a replacement at any time the device Hostname will change. Please consider the implications to the NetworkHD API communications when doing this.

The format of an endpoint Hostname is as follows:

<Device Type>-<Device MAC Address>
e.g.: NHD-500-TX-E4CE02F34D67

A device Alias is a name given to a NetworkHD endpoint using the NetworkHD Console software utility included within the WyreStorm Management Suite software package. It is advised that during NetworkHD initial setup a suitable Alias is given to a NetworkHD endpoint. Using a device Alias within the API communications encourages the flexibility to define a set reference to a particular endpoint location, so that physical encoders or decoders may be swapped out for a different device when required but still maintain the same API reference.

The Alias reference for a NetworkHD endpoint is normally configured during initial setup but can be manually changed using the NetworkHD Console software utility at any time, if required. Please consider the implications to the NetworkHD API communications when doing this.

The 3rd-party controller can use either Hostname or Alias references for NetworkHD endpoints in a command sent to the NHD-CTL.

The NHD-CTL, when using factory default settings, will use Alias mode for API command responses and API notifications.

## 3.8   Command Structure Key Elements

| Command Element | Description |
|---|---|
| <A> | Variable placeholder where A is the name of the variable. Variable data options are noted in the command table. Only one of the data options can be used in a command. |
| <A\|B> | Variable placeholder listing all the variable name options (A or B), separated by the pipe (OR) symbol. Variable data options are noted in the command table. Only one of the data options can be used in a command. |
| [Y\|Z] | Command data placeholder listing all the data options directly (Y or Z), separated by the pipe (OR) symbol. Only one of the data options can be used in a command. |
| (<A>) | Variable placeholder where the use of data is optional, depending on the context of the command |
| ([Y]) | Command data placeholder where the use of data is optional, depending on the context of the command. |
| ... | Signifies optional continuation of data following the structure of the defined command. |

# 4.  API Endpoint Reference

## 4.1  Session Alias Mode

For the API session (Telnet or RS-232), the command responses or notifications from the NHD-CTL can reference either the endpoint Hostname or Alias. For each new Telnet session, or after the NHD-CTL is rebooted, the default mode is applied.

| Configure Alias mode for API responses or notifications from NHD-CTL | |
| --- | --- |
| Command structure:<br>`config set session alias [on\|off]` | **on** = API responses and notifications will reference the endpoint by Alias |
| Response structure (command mirror):<br>`config set session alias [on\|off]` | **off** = API responses and notifications will reference the endpoint by hostname |
| Command Example: Session to use hostname referencing<br>`config set session alias off` | |
| Response Example: Command acknowledgment<br>`config set session alias off` | New session default: **on** |
| Command applies to:<br>NHD-000-CTL, NHD-CTL-PRO | |

# 5.     Reboot/Reset Commands

## 5.1  Device Reboot

To reboot a single or multiple NetworkHD devices (retains current settings):

| Reboot the NHD-CTL | |
|---|---|
| Command structure:<br>**config set reboot** | |
| Response structure:<br>**system will reboot now** | |
| Command Example: Reboot the NHD-CTL<br>`config set reboot` | |
| Response Example: Command acknowledgment<br>`system will reboot now` | |
| Command applies to:<br>NHD-000-CTL, NHD-CTL-PRO | |

| Reboot TX or RX devices | |
|---|---|
| Command structure:<br>**config set device reboot <TX1\|RX1> <TX\|RX2>** | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>**the following device will reboot now:<CR><LF> <TX1\|RX1><CR><LF><TX2\|RX2>** | **RXn** = Decoder reference – Alias or hostname |
| Command Example: Reboot encoders<br>`config set device reboot source1 source2` | **CR** = Carriage Return byte (0x0D) (CR) (\r) |
| Response Example: Command acknowledgment<br>`the following device will reboot now:`<br>`source1`<br>`source2` | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

## 5.2  Device Reset

To rest a single or multiple NetworkHD devices (defaults all settings):

| Reset TX or RX devices | |
|---|---|
| Command structure:<br>**config set device restorefactory <TX1\|RX1> <TX\|RX2>** | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>**the following device will restore now:<CR><LF> <TX1\|RX1><CR><LF><TX2\|RX2> … <CR><LF><TXn\|RXn>** | **RXn** = Decoder reference – Alias or hostname |
| Command Example: Reboot encoders<br>`config set device restorefactory source1 source2` | **CR** = Carriage Return byte (0x0D) (CR) (\r) |
| Response Example: Command acknowledgment<br>`the following device will restore now:`<br>`source1`<br>`source2` | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

# 6.  Media Stream Matrix Switch Commands

## 6.1  Stream Matrix Switching – All Media

Use this command to switch all primary media streams available to the NetworkHD endpoints simultaneously. Where available at the NetworkHD endpoints, this will include the switching at the same time of primary video, primary audio, RS-232, Infrared and USB between the TX and the RX.

| Assign all primary TX media streams to RX(s) | |
|---|---|
| Command structure:<br>`matrix set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix set <TX1> <RX1> <RX2> … <RXn>` | |
| Command Example: Assign encoder to decoders<br>`matrix set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix set source1 display1 display2 display3 display4` | |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |
| Notes:<br>`matrix set` commands do not apply to Multiview, only decoders. See section 11- Multiview | |

| Unassign all primary media streams from RX | |
|---|---|
| Command structure:<br>`matrix set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign all media streams from RXs<br>`matrix set null display1 display2` | |
| Response Example: Command acknowledgment<br>`matrix set null display1 display2` | |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |
| Notes:<br>`matrix set` commands do not apply to Multiview, only decoders. See section 11- Multiview. | |

## 6.2  Stream Matrix Switching – Video Stream Breakaway

Use this command to switch video streams available to the NetworkHD endpoints independently. Commands will switch the video streams only and leave the current assignment for all other media streams unaffected.

| Assign RX(s) video output to TX stream | |
|---|---|
| Command structure:<br>`matrix video set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix video set <TX1> <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Command Example: Assign encoder video stream to decoders<br>`matrix video set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix video set source1 display1 display2 display3 display4` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |
| Notes:<br>`matrix video set` commands do not apply to Multiview, only decoders. See section 11- Multiview. | |

| Unassign video stream from RX | |
|---|---|
| Command structure:<br>`matrix video set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix video set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign video stream from RXs<br>`matrix video set null display1 display2` | |
| Response Example: Command acknowledgment<br>`matrix video set null display1 display2` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |
| Notes:<br>`matrix video set` commands do not apply to Multiview, only decoders. See section 11- Multiview. | |

## 6.3 Stream Matrix Switching – Audio Stream Breakaway

Use this command to switch audio streams available to the NetworkHD endpoints independently. Commands will switch the audio streams only and leave the current assignment for all other media streams unaffected.

| Assign RX(s) HDMI audio output to TX stream | |
|---|---|
| Command structure:<br>`matrix audio set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix audio set <TX1> <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Command Example: Assign encoder audio stream to decoders<br>`matrix audio set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix audio set source1 display1 display2 display3 display4` | |

Command applies to:
NHD-110-TX, NHD-110-RX
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Analog audio output on a 100, 200 and 400 Series decoder will extract the HDMI audio at the decoder.
HDMI audio on a 600 Series decoder will select the TX audio stream defined using the `config set device audiosource` command – see section 7.2 – Port Switching – Audio. The primary HDMI or DP audio stream is the factory default setting.
Analog audio output on a 600 Series decoder will output the TX primary HDMI or DP audio stream if "dmix" is set using the `config set audio2source` command – see section 7.2 – Port Switching – Audio.
Analog audio output on a 100, 200 and 400 Series decoder will convert 2 channel PCM audio only.
Analog audio output on a 600 Series decoder can also downmix up to 8 channel PCM audio.
`matrix audio set` commands do not apply to Multiview, only decoders. See section 11- Multiview.

| Unassign RX(s) HDMI audio output to TX stream | |
|---|---|
| Command structure:<br>`matrix audio set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix audio set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign encoder audio stream to decoders<br>`matrix audio set null display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix audio set null display1 display2 display3 display4` | |

Command applies to:
NHD-110-RX, NHD-110-RX
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Analog audio output on a 100, 200 and 400 Series decoder will extract the HDMI audio at the decoder.
HDMI audio on a 600 Series decoder will select the TX audio stream defined using the `config set device audiosource` command – see section 7.2 – Port Switching – Audio. The primary HDMI or DP audio stream is the factory default setting.
Analog audio output on a 600 Series decoder will output the TX primary HDMI or DP audio stream if "dmix" is set using the `config set audio2source` command – see section 7.2 – Port Switching – Audio.
`matrix audio set` commands do not apply to Multiview, only decoders. See section 11- Multiview.

## Assign RX(s) analog audio output to TX analog audio stream

| Command structure:<br>`matrix audio2 set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| --- | --- |
| Response structure (command mirror):<br>`matrix audio2 set <TX1> <RX1> <RX2> … <RXn>` | |
| Command Example: Assign encoder audio stream to decoders<br>`matrix audio2 set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix audio2 set source1 display1 display2 display3 display4` | |

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Command will only be effective if "analog" is set using the `config set audio2source` command – see section 7.2 – Port Switching – Audio. This is the factory default setting.


## Assign RX(s) ARC stream to TX

| Command structure:<br>**matrix audio3 set <RX1> <TX1>** | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| --- | --- |
| Response structure (command mirror):<br>**matrix audio3 set <RX1> <TX1>** | |
| Command Example: Assign decoder ARC stream to encoder<br>**matrix audio3 set Display1 Source1** | |
| Response Example: Return data<br>**matrix audio3 set Display1 Source1** | |

Command applies to:
NHD-500-TX, NHD-500-RX


## Unassign RX(s) analog audio output to TX analog audio stream

| Command structure:<br>`matrix audio2 set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| --- | --- |
| Response structure (command mirror):<br>`matrix audio2 set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign encoder audio stream to decoders<br>`matrix audio2 set null display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix audio2 set null display1 display2 display3 display4` | |

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Command will only be effective if "analog" is set using the `config set audio2source` command – see section 7.2 – Port Switching – Audio. This is the factory default setting.

North America: 844.280.WYRE (9973) | UK/EMEA: +44 (0) 1793 230 343
Email: Support@WyreStorm.com

## 6.4 Stream Matrix Switching – USB Stream Breakaway

Use this command to switch USB streams available to the NetworkHD endpoints independently. Commands will switch the USB streams only and leave the current assignment for all other media streams unaffected.

| Assign RX(s) USB ports to TX | |
|---|---|
| Command structure:<br>`matrix usb set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix usb set <TX1> <RX1> <RX2> … <RXn>` | |
| Command Example: Assign encoder USB port to decoders<br>`matrix usb set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix usb set source1 display1 display2 display3 display4` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

Notes:
- NHD-600 devices only support a 1:1 USB communication, meaning only one RX can be linked to a TX USB host port.
- The NHD-500-RX "USB 2.0" port only supports a 1:1 USB communication, meaning only one RX's USB 2.0 port can be linked to a TX USB host port.
- Command will always be effective for RX USB ports in KMoIP mode.
- Command will only be effective for RX USB ports in USBoIP mode if the TX has 4 or less USBoIP mode RX ports already assigned.
- To assign an endpoint USB mode – see section 7.3 Port Switching – USB Mode

| Unassign RX(s) USB ports | |
|---|---|
| Command structure:<br>`matrix usb set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix usb set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign decoder usb ports<br>`matrix usb set null display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix usb set null display1 display2 display3 display4` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

## 6.5 Stream Matrix Switching – Infrared Stream Breakaway

Use this command to switch infrared streams available to the NetworkHD endpoints independently. Commands will switch the infrared streams only and leave the current assignment for all other media streams unaffected.

| Assign RX(s) infrared ports to TX | |
|---|---|
| Command structure:<br>`matrix infrared set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix infrared set <TX1> <RX1> <RX2> … <RXn>` | |
| Command Example: Assign encoder IR ports to decoder IR ports<br>`matrix infrared set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix infrared set source1 display1 display2 display3 display4` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| Notes:<br>Encoder to decoder(s) assignments link the IR channels in both directions - the IR receiver port on the encoder to the IR emitter port on the decoder simultaneously with the IR receiver port on the decoder to the IR emitter port on the encoder.<br>To generate IR codes directly from an encoder or decoder using IR code Hex notation, see section 8.2 - Device Control – Custom Command Generation. | |

| Unassign RX(s) infrared ports (infrared set) | |
|---|---|
| Command structure:<br>`matrix infrared set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix infrared set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign decoder IR ports<br>`matrix infrared set null display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix infrared set null display1 display2 display3 display4` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |

North America: 844.280.WYRE (9973) | UK/EMEA: +44 (0) 1793 230 343
Email: Support@WyreStorm.com

## Assign RX or TX infrared receiver port to endpoint(s) or API notification

Command structure:
```
matrix infrared2 set <TX|RX> <mode> (<TX|RX>)
```

Response structure (command mirror):
```
matrix infrared2 set <TX|RX> <mode> (<TX|RX>)
```

Command Example 1: Assign encoder IR receiver port to decoder IR emitter port
```
matrix infrared2 set source1 single display1
```

Response Example: Command acknowledgment
```
matrix infrared2 set source1 single display1
```

Command Example 2: Assign encoder IR receiver port to encoder IR emitter port
```
matrix infrared2 set source1 single source2
```

Response Example: Command acknowledgment
```
matrix infrared2 set source1 single source2
```

Command Example 3: Assign decoder IR receiver port to API notification
```
matrix infrared2 set display1 api
```

Response Example: Command acknowledgment
```
matrix infrared2 set display1 api
```

Command Example 4: Assign encoder IR receiver port to all endpoint IR emitter ports
```
matrix infrared2 set source1 all
```

Response Example: Command acknowledgment
```
matrix infrared2 set source1 all
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**mode** = Operational mode:
- **single** = Assign to endpoint
- **api** = Assign to API notification
- **all** = Assign to all other endpoints

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Assignments are in one direction, from endpoint IR receiver port to endpoint IR emitter port or API notification.
For details on infrared API notifications, see section 12.2 – Endpoint Notifications.
To generate IR codes directly from an encoder or decoder using IR code Hex notation, see section 8.2 - Device Control – Custom Command Generation.

## Unassign TX or RX infrared receiver port (infrared2 set)

Command structure:
```
matrix infrared2 set <TX|RX> null
```

Response structure (command mirror):
```
matrix infrared2 set <TX|RX> null
```

Command Example: Unassign decoder IR receiver port
```
matrix infrared2 set display1 null
```

Response Example: Command acknowledgment
```
matrix infrared2 set display1 null
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

## 6.6 Stream Matrix Switching – RS-232 Stream Breakaway

Use this command to switch RS-232 streams available to the NetworkHD endpoints independently. Commands will switch the RS-232 streams only and leave the current assignment for all other media streams unaffected.

| Assign RX(s) RS-232 ports to TX | |
| --- | --- |
| Command structure:<br>`matrix serial set <TX1> <RX1> <RX2> … <RXn>` | **TXn** = Encoder reference – Alias or hostname<br><br>**RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix serial set <TX1> <RX1> <RX2> … <RXn>` | |
| Command Example: Assign encoder RS-232 port to decoders<br>`matrix serial set source1 display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix serial set source1 display1 display2 display3 display4` | |
| Command applies to:<br>NHD-400-TX-IW, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| Notes:<br>The RS-232 stream is a bidirectional channel.<br>To generate RS-232 codes directly from an encoder or decoder using ASCII or Hex notation, see section 8.2 - Device Control – Custom Command Generation. | |

| Unassign RX(s) RS-232 ports (serial set) | |
| --- | --- |
| Command structure:<br>`matrix serial set null <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`matrix serial set null <RX1> <RX2> … <RXn>` | |
| Command Example: Unassign decoder RS-232 ports<br>`matrix serial set null display1 display2 display3 display4` | |
| Response Example: Command acknowledgment<br>`matrix serial set null display1 display2 display3 display4` | |
| Command applies to:<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |

## Assign RX or TX RS-232 port to endpoint(s) or API notification

Command structure:
```
matrix serial2 set <TX|RX> <mode> (<TX|RX>)
```

Response structure (command mirror):
```
matrix serial2 set <TX|RX> <mode> (<TX|RX>)
```

Command Example 1: Assign encoder RS-232 port to decoder
```
matrix serial2 set source1 single display1
```

Response Example: Command acknowledgment
```
matrix serial2 set source1 single display1
```

Command Example 2: Assign encoder RS-232 port to encoder
```
matrix serial2 set source1 single source2
```

Response Example: Command acknowledgment
```
matrix serial2 set source1 single source2
```

Command Example 3: Assign decoder RS-232 port to API notification
```
matrix serial2 set display1 api
```

Response Example: Command acknowledgment
```
matrix serial2 set display1 api
```

Command Example 4: Assign encoder RS-232 port to all endpoint RS-232 ports
```
matrix serial2 set source1 all
```

Response Example: Command acknowledgment
```
matrix serial2 set source1 all
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**mode** = Operational mode:
 **single** = Assign to endpoint
 **api** = Assign to API notification
 **all** = Assign to all other endpoints

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The RS-232 stream is a bidirectional channel.
To generate RS-232 codes directly from an encoder or decoder using ASCII or Hex notation, see section 8.2 - Device Control – Custom Command Generation.

## Unassign TX or RX RS-232 port (serial2 set)

Command structure:
```
matrix serial2 set <TX|RX> null
```

Response structure (command mirror):
```
matrix serial2 set <TX|RX> null
```

Command Example: Unassign decoder RS-232 port
```
matrix serial2 set display1 null
```

Response Example: Command acknowledgment
```
matrix serial2 set display1 null
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

# 7.    Device Port Switch Commands

## 7.1  Port Switching – Video

Use this command to define which video source input port an encoder will use. Where available at the NetworkHD endpoints, this will include the switching between discrete video source input ports to select which connected video source will be used with encoder AV over IP processing.

| Switch TX video input port | |
|---|---|
| Command structure:<br>`config set device videosource <TX> <source>` | **TX** = Encoder reference – Alias or hostname<br><br>**source** = Video input port:<br>    **auto** = Automatic selection (default)<br>    **hdmi** = HDMI port<br>    **dp** = DisplayPort port |
| Response structure (command mirror):<br>`config set device videosource <TX> <source>` | |
| Command Example: Select the HDMI port at the encoder<br>`config set device videosource source1 hdmi` | |
| Response Example: Command acknowledgment<br>`config set device videosource source1 hdmi` | |
| Command applies to:<br>NHD-600-TX | |

## 7.2 Port Switching – Audio

Use this command to define which audio source a decoder port will use when assigned to a specific encoder. Where multiple audio sources are available to a NetworkHD decoder, this will include the switching between discrete audio sources to select which audio source will be used with a decoder port.

| Switch RX HDMI audio output port to a discrete audio stream from an assigned TX | |
|---|---|
| Command structure:<br>`config set device audiosource <RX> <source>` | **RX** = Decoder reference – Alias or hostname<br><br>**source** = Assigned encoder's audio stream:<br>    **hdmi** = HDMI audio stream (default)<br>    **dmix** = 2ch downmix HDMI audio stream<br>    **analog** = Analog port stream |
| Response structure (command mirror):<br>`config set device audiosource <RX> <source>` | |
| Command Example: Select the assigned encoder's HDMI audio stream<br>`config set device audiosource display1 hdmi` | |
| Response Example: Command acknowledgment<br>`config set device audiosource display1 hdmi` | |
| Command applies to:<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Where an encoder offers multiple audio streams, a decoder will switch to the audio stream on that encoder defined by this command. This remains the stream selection when assigning a different encoder using the Media Stream Matrix Switch commands found in section 6 – Media Stream Matrix Switch Commands.

## Switch RX analog audio output port to a discrete audio source

| | |
|---|---|
| Command structure:<br>`config set device audio2source <RX> <stream>` | **RX** = Decoder reference – Alias or hostname<br><br>**source** = Audio source:<br>    **analog** = Assigned encoder's analog<br>          audio stream (default)<br>    **dmix** = 2ch downmix of the RX HDMI<br>          audio output |
| Response structure (command mirror):<br>`config set device audio2source <RX> <stream>` | |
| Command Example: Select a downmix of the decoder's HDMI audio output<br>`config set device audio2source display1 dmix` | |
| Response Example: Command acknowledgment<br>`config set device audio2source display1 dmix` | |

Command applies to:
NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Where `analog` is selected, the decoder analog audio output port joins with the assigned encoder analog audio input port. This remains the stream selection and will assign to a different encoder when using the `matrix set` commands found in section 6.1 – Stream Matrix Switching – All Media or the `matrix audio2 set` commands found in section 6.2 – Stream Matrix Switching – Audio Stream Breakaway. The decoder analog output port will NOT assign to a different encoder when using the discrete HDMI audio `matrix audio set` command, found in section 6.2 – Stream Matrix Switching – Audio Stream Breakaway.
Where `dmix` is selected, the decoder analog audio output port always follows the decoder HDMI audio output.
The decoder analog audio output port offers 2 channel downmixing of up to 8 channel PCM audio.


## Switch audio input port on NHD-500-DNT-TX for main audio stream

| | |
|---|---|
| Command structure:<br>`config set device audio input type <source> <TX>` | **TX** = Encoder reference – Alias or hostname<br><br>**source** = Video input port:<br>    **auto** = Automatic selection (default)<br>    **hdmi** = HDMI port<br>    **analog** = 3.5mm port |
| Response structure (command mirror):<br>`config set device audio input type <source> <TX>` | |
| Command Example: Select the HDMI port at the encoder<br>`config set device audio input type hdmi source1` | |
| Response Example: Command acknowledgment<br>`config set device audio input type hdmi source1` | |

Command applies to:
NHD-500-DNT-TX


## Switch audio input port on NHD-500-DNT-TX for Dante stream

| | |
|---|---|
| Command structure:<br>`config set device info dante.audio_input=<source> <TX>` | **TX** = Encoder reference – Alias or hostname<br><br>**source** = Video input port:<br>    **hdmi** = HDMI port (default)<br>    **analog** = 3.5mm port |
| Response structure (command mirror):<br>`config set device info dante.audio_input=<source> <TX>` | |
| Command Example: Select the HDMI port at the encoder<br>`config set device info dante.audio_input=hdmi source1` | |
| Response Example: Command acknowledgment<br>`config set device info dante.audio_input=hdmi source1` | |

Command applies to:
NHD-500-DNT-TX

## 7.3 Port Switching – USB Mode

Use this command to define the working mode for a USB port. Where available to a NetworkHD endpoint, this will configure the working mode of the USB ports on an endpoint.

| Configure USB working mode for endpoint | |
|---|---|
| Command structure:<br>`config set device info km_over_ip_enable=<on|off> <TX|RX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure (command mirror):<br>`config set device info km_over_ip_enable=<on|off> <TX|RX>` | **RX** = Decoder reference – Alias or hostname |
| Command Example: Disable automatic KMoIP mode on endpoint<br>`config set device info km_over_ip_enable=off display1` | **on** = Automatic KMoIP mode enabled (default) |
| Response Example: Command acknowledgment<br>`config set device info km_over_ip_enable=off display1` | **off** = KMoIP mode disabled – Use USBoIP mode |
| Command applies to:<br>NHD-400-TX, NHD-400-RX | |

Notes:
When automatic KMoIP mode is enabled on an RX, this mode will be used when the RX detects a connected device that uses a USB HID Class (Base Class 03).
When automatic KMoIP mode is enabled on a TX, this will prompt a connected device to enumerate a composite HID device.
For more detail on the USB operating modes, please refer to the NetworkHD Technical Reference Guide.

# 8.   Connected Device Control

## 8.1  Device Control – Proxy Commands

Use this command to generate preset control commands for connected devices. Where available at the NetworkHD endpoints, this will include sending CEC and/or RS-232 data to a connected device.

Proxy commands are defined using NetworkHD Console, part of the WyreStorm Management Suite. Each decoder can have different command data that is sent to a connected device when the proxy command is received.

| Send "power" proxy command for device attached to RX | |
|---|---|
| Command structure:<br>`config set device sinkpower [on\|off] <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`config set device sinkpower [on\|off] <RX1> <RX2> … <RXn>` | **on** = Power on proxy command<br><br>**off** = Power off proxy command |
| Command Example: Instruct decoder to send its unique power on command<br>`config set device sinkpower on display1` | |
| Response Example: Command acknowledgment<br>`config set device sinkpower on display1` | |
| Command applies to: | |
| NHD-110-RX, NHD-100-RX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX | |
| NHD-400-RX, NHD-500-RX | |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Power on and power off commands are defined as either CEC, RS-232 or both within NetworkHD Console. By default, power commands are disabled on each endpoint. Power commands can be specified uniquely for each endpoint.

| Send CEC proxy command for device attached to RX | |
|---|---|
| Command structure:<br>`config set device cec [onetouchplay\|standby] <RX1> <RX2> … <RXn>` | **RXn** = Decoder reference – Alias or hostname |
| Response structure (command mirror):<br>`config set device cec [onetouchplay\|standby] <RX1> <RX2> … <RXn>` | **onetouchplay** = CEC power on proxy command |
| Command Example: Instruct decoder to send its unique CEC standby command<br>`config set device cec standby display1` | **standby** = CEC power off proxy command |
| Response Example: Command acknowledgment<br>`config set device cec standby display1` | |
| Command applies to: | |
| NHD-110-RX, NHD-100-RX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX | |
| NHD-400-RX, NHD-500-RX | |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
CEC power on and power off commands are defined within the NetworkHD Console. By default, CEC power commands will use the default CEC data commands. For details on the default CEC commands, see the NetworkHD Technical Reference Guide.
Custom CEC power on and power off commands can be set within the NetworkHD Console to replace the default CEC commands.
Unlike `config set device sinkpower`, the `config set device cec` command cannot be disabled in the NetworkHD Console and will always generate data from the decoder.

## 8.2 Device Control – Custom Command Generation

Use this command to generate custom control commands for connected devices. Where available at the NetworkHD endpoints, this will include sending custom command data to a connected device directly from the endpoint.

| Send custom CEC command to device attached to RX | |
|---|---|
| Command structure:<br>`cec "<cecdata>" <RX>` | **RX** = Decoder reference – Alias or hostname<br><br>**cecdata** = CEC data bytes in hex format |
| Response structure (command mirror):<br>`cec "<cecdata>" <RX>` | |
| Command Example: Instruct decoder to send custom CEC command | |
| `cec "FF36" display1` | |
| Response Example: Command acknowledgment | |
| `cec "FF36" display1` | |
| Command applies to: | |
| NHD-110-RX, NHD-100-RX, NHD-200-RX, NHD-210-RX, NHD-250-RX | |
| NHD-400-RX, NHD-500-RX | |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
CEC data bytes are in hexadecimal notation with no space separation between bytes.

| Send custom infrared command to device attached to RX or TX | |
|---|---|
| Command structure:<br>`infrared "<irdata>" <TX|RX>` | **TX** = Encoder reference – Alias or hostname<br><br>**RX** = Decoder reference – Alias or hostname<br><br>**irdata** = Infrared command data |
| Response structure (command mirror):<br>`infrared "<irdata>" <TX|RX>` | |
| Command Example: Instruct encoder to send custom IR command | |
| `infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030`<br>`0018 0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018`<br>`0018 0018 0030 0018 0030 0018 0018 0018 0030 0018 0018 0018 0018`<br>`0018 0018 0018 0030 0018 0030 0018 0030 01FE" source1` | |
| Response Example: Command acknowledgment | |
| `infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030`<br>`0018 0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018`<br>`0018 0018 0030 0018 0030 0018 0018 0018 0030 0018 0018 0018 0018`<br>`0018 0018 0018 0030 0018 0030 0018 0030 01FE" source1` | |
| Command applies to: | |
| NHD-110-TX, NHD-110-RX | |
| NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |
| NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

Notes:
NHD-400 & 600 series require infrared data in Pronto CCF Hex format.

NHD-110 series FW v6.0.xx requires infrared data in Global Cache format. FW v7.0.xx requires Pronto Hex format.

## Send custom RS-232 command to device attached to RX or TX

Command structure:
```
serial -b <baud>-<bits><parity><stop> -r <CR> -n <LF> -h <HEX>
"<data>" <TX|RX>
```

Response structure (command mirror):
```
serial -b <baud>-<bits><parity><stop> -r <CR> -n <LF> -h <HEX>
"<data>" <TX|RX>
```

Command Example 1: Instruct encoder to send custom RS-232 ASCII command
```
serial -b 115200-8n1 -r on -n on -h off "vol +80dB" source1
```

Response Example: Command acknowledgment
```
serial -b 115200-8n1 -r on -n on -h off "vol +80dB" source1
```

Command Example 2: Instruct decoder to send custom RS-232 Hex command
```
serial -b 19200-7e2 -r off -n off -h on "4c 6f 72 65 6d 20 69 70
73 75 6d 20 64 6f 6c 6f 72 20 73 69 74 20 61 6d 65 74 2c 20 63 6f
6e 73 65 63 74 65 74 75 72 20 61 64 69 70 69 73 63 69 6e 67 20 65
6c 69 74 2c 20 73 65 64 20 64 6f 20 65 69 75 73 6d 6f 64 20 74 65
6d 70 6f 72 20 69 6e 63 69 64 69 64 75 6e 74 20 75 74 20 6c 61 62
6f 72 65 20 65 74 20 64 6f 6c 6f 72 65 20 6d 61 67 6e 61 20 61 6c
69 71 75 61 0D 0A" display1
```

Response Example: Command acknowledgment
```
serial -b 19200-7e2 -r off -n off -h on "4c 6f 72 65 6d 20 69 70
73 75 6d 20 64 6f 6c 6f 72 20 73 69 74 20 61 6d 65 74 2c 20 63 6f
6e 73 65 63 74 65 74 75 72 20 61 64 69 70 69 73 63 69 6e 67 20 65
6c 69 74 2c 20 73 65 64 20 64 6f 20 65 69 75 73 6d 6f 64 20 74 65
6d 70 6f 72 20 69 6e 63 69 64 69 64 75 6e 74 20 75 74 20 6c 61 62
6f 72 65 20 65 74 20 64 6f 6c 6f 72 65 20 6d 61 67 6e 61 20 61 6c
69 71 75 61 0D 0A" display1
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**baud** = Baud rate:
> **2400 4800 9600 19200 38400
> 57600 115200**

**bits** = Number of data bits:
> **6 7 8**

**parity** = Parity bit odd / even / none:
> **o e n**

**stop** = Number of stop bits:
> **1 2**

**CR** = Append carriage return delimiter (\r)(CR):
> **on off**

**LF** = Append line feed delimiter (\n)(LF):
> **on off**

**HEX** = Data format:
> **on** = Hexadecimal format
> **off** = ASCII format

**data** = RS-232 command data

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
Where specified, delimiters are appended to the end of the transmitted data.
Delimiters cannot be included in command data (data) when using ASCII format e.g. \r or \n are not escaped and sent literally as ASCII.
All hex data byte values are supported when using hexadecimal command data format (00 through FF) and are space-separated.

# 9.   Audio Output

## 9.1  Volume Control – Analog Audio

Use this command to alter the audio volume level of an endpoint analog audio output. Where available at the NetworkHD endpoints, this will alter the audio volume exiting an analog audio port.

| Alter the analog audio volume from a TX or RX | |
|---|---|
| Command structure:<br>`config set device audio volume <level> analog <TX\|RX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure (command mirror):<br>`config set device audio volume <level> analog <TX\|RX>` | **RX** = Decoder reference – Alias or hostname<br><br>`level` = volume command |
| Command Example: Decrease the audio volume level from decoder<br>`config set device audio volume down analog display1` |       `up`<br>      `down` |
| Response Example: Command acknowledgment<br>`config set device audio volume down analog display1` |       `mute`<br>      `unmute` |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX<br>NHD-400-TX, NHD-400-RX | |

Notes:
NetworkHD 400 Series endpoints only support volume up and volume down commands and not the mute and unmute commands.

NetworkHD 100/200 Series endpoints only support mute and unmute and not the volume up and down commands

The factory default volume level for a NetworkHD endpoint is maximum volume level (unmuted).

# 10.	Video Wall

## 10.1 Video Wall – 'Standard Video Wall' Scenes

Use this command to apply preset 'standard video wall' Scenes to a video wall. Where available at the NetworkHD endpoints, 'standard video wall' scenes configured in the NetworkHD Console software utility can be recalled and applied to physical video wall arrays.

| Apply a 'standard video wall' Scene to a physical video wall array | |
|---|---|
| Command structure:<br>`scene active <videowall>-<scene>` | `videowall` = Preconfigured video wall name<br><br>`scene` = Preconfigured Scene name |
| Response structure:<br>`scene <videowall>-<scene> active [success|failure]` | |
| Command Example: Apply a videowall scene<br>`scene active OfficeVW-Splitmode` | |
| Response Example: Command acknowledgment<br>`scene OfficeVW-Splitmode active success` | |
| Command applies to:<br>NHD-110-RX, NHD-210-RX<br>NHD-400-RX, NHD-500-RX<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Video wall names are given to physical video wall arrays when created in the NetworkHD Console software utility.
When video wall arrays are created in Console, the physical displays will have decoders assigned to them.
Scene names are given to Scenes when created in the NetworkHD Console software utility.
When scenes are created in Console, the decoders will have default encoders or no encoder (NULL) assigned to them.

| Change the encoder assigned to a single decoder in a 'standard video wall' Scene | |
|---|---|
| Command structure:<br>`scene set <videowall>-<scene> <X> <Y> <TX>` | `TX` = Encoder reference – Alias or hostname<br><br>`videowall` = Preconfigured video wall name |
| Response structure:<br>`scene <videowall>-<scene>'s source in [<X>,<Y>] change to <TX>` | `scene` = Preconfigured Scene name |
| Command Example: Change the encoder assigned to decoder in a scene<br>`scene set OfficeVW-Splitmode 1 2 source1` | `X` = Decoder's horizontal position in display array |
| Response Example: Command acknowledgment<br>`scene OfficeVW-Splitmode's source in [1,2] change to source1` | `Y` = Decoder's vertical position in display array |
| Command applies to:<br>NHD-110-RX, NHD-210-RX<br>NHD-400-RX, NHD-500-RX<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Command commits and saves change to Scene but does not apply live until using the `scene active` command.
Command will only be effective for decoders that are not part of a Logical Screen element in a Scene.
Horizontal and vertical positions are relative to the top left display.

| Change the encoder assigned to a Logical Screen in a 'standard video wall' Scene | |
|---|---|
| Command structure:<br>`vw change <videowall>-<scene>_<Lscreen> <TX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure:<br>`videowall <videowall>-<scene>_<Lscreen> tx connect to <TX>` | **videowall** = Preconfigured video wall name |
| Command Example: Change the encoder assigned to a Logical Screen in a scene<br>`vw change OfficeVW-Combined_TopTwo source1` | **scene** = Preconfigured Scene name<br><br>**Lscreen** = Preconfigured Logical Screen name |
| Response Example: Command acknowledgment<br>`videowall change OfficeVW-Combined_TopTwo tx connect to source1` | |

Command applies to:

NHD-110-RX, NHD-210-RX

NHD-400-RX, NHD-500-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Command will be effective immediately for the Logical Screen and will not save to Scene.
Following with a `scene active` command will reset the Logical Screen to the default encoder for the Scene.

## 10.2 Video Wall – 'Video Wall within a Wall' Scenes

Use this command to apply window functions to 'video wall within a wall' Scenes to a video wall. Where available at the NetworkHD endpoints, 'videowall within a wall' scenes configured in the NetworkHD Console software utility can be recalled and manipulation of windows applied to physical video wall arrays.

| Apply a 'video wall within a wall' Scene to a physical video wall array | |
|---|---|
| Command structure:<br>`wscene2 active <videowall>-<wscene>` | `videowall` = Preconfigured video wall name |
| Response structure:<br>`wscene2 active <videowall>-<wscene> [success\|failure]` | `wscene` = Preconfigured Scene name |
| Command Example: Apply a video wall scene<br>`wscene2 active OfficeVW-windowscene` | |
| Response Example: Command acknowledgment<br>`wscene2 active OfficeVW-windowscene success` | |
| Command applies to:<br>NHD-110-RX, NHD-210-RX<br>NHD-400-RX, NHD-500-RX<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Video wall names are given to physical video wall arrays when created in the NetworkHD Console software utility.
When video wall arrays are created in the NetworkHD Console, the physical displays will have decoders assigned to them.
Scene names are given to Scenes when created in the NetworkHD Console software utility.
When scenes are created in the NetworkHD Console, the decoders will have no encoder (NULL) assigned to them.

| Open a window within a 'video wall within a wall' Scene | |
|---|---|
| Command structure:<br>`wscene2 window open <videowall>-<wscene> <wname> <X> <Y> <H> <V> <TX>` | `TX` = Encoder reference – Alias or hostname |
| | `videowall` = Preconfigured video wall name |
| Response structure:<br>`wscene2 window open <videowall>-<wscene> <wname> <X> <Y> <H> <V> <TX> [success\|failure]` | `wscene` = Preconfigured Scene name |
| | `wname` = Unique window name |
| Command Example: Open a window within a scene<br>`wscene2 window open OfficeVW-windowscene window1 0 0 2 2 source1` | `X` = Window's top left decoder horizontal position in display array |
| | `Y` = Window's top left decoder vertical position in display array |
| Response Example: Command acknowledgment<br>`wscene2 window open OfficeVW-windowscene window1 0 0 2 2 source1 success` | `H` = Window's horizontal size in number of displays |
| | `V` = Window's vertical size in number of displays |
| Command applies to:<br>NHD-110-RX, NHD-210-RX<br>NHD-400-RX, NHD-500-RX<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Positions are ZERO INDEXED in this command, e.g. top-left display is X = 0, Y = 0
Command is effective immediately and does not save to the default Scene.
Horizontal and vertical positions are relative to the top left display.
Window names must be unique to the command and are not defined using the NetworkHD Console software utility.

| Close a window within a 'video wall within a wall' Scene | |
|---|---|
| Command structure:<br>`wscene2 window close <videowall>-<wscene> <wname>` | `videowall` = Preconfigured video wall name |
| Response structure:<br>`wscene2 window close <videowall>-<wscene> <wname>`<br>`[success\|failure]` | `wscene` = Preconfigured Scene name<br><br>`wname` = Unique window name |
| Command Example: Close a window within a scene<br>`wscene2 window close OfficeVW-windowscene window1` | |
| Response Example: Command acknowledgment<br>`wscene2 window close OfficeVW-windowscene window1 success` | |

Command applies to:

| |
|---|
| NHD-110-RX, NHD-210-RX |
| NHD-400-RX, NHD-500-RX |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX |

Notes:
Command is effective immediately and does not save to the default Scene.

| Change the TX for an open window within a 'video wall within a wall' Scene | |
|---|---|
| Command structure:<br>`wscene2 window change <videowall>-<wscene> <wname> <TX>` | `TX` = Encoder reference – Alias or hostname |
| Response structure:<br>`wscene2 window change <videowall>-<wscene> <wname> <TX>`<br>`[success\|failure]` | `videowall` = Preconfigured video wall name<br><br>`wscene` = Preconfigured Scene name<br><br>`wname` = Unique window name |
| Command Example: Reposition and resize a window within a scene<br>`wscene2 window change OfficeVW-windowscene window1 source1` | |
| Response Example: Command acknowledgment<br>`wscene2 window change OfficeVW-windowscene window1 source1`<br>`success` | |

Command applies to:

| |
|---|
| NHD-110-RX, NHD-210-RX |
| NHD-400-RX, NHD-500-RX |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX |

Notes:
Command is effective immediately and does not save to the default Scene.
Window names must use names of windows currently open in the Scene.

## Reposition and resize an open window within a 'video wall within a wall' Scene

Command structure:
`wscene2 window adjust <videowall>-<wscene> <wname> <X> <Y> <H> <W>`

Response structure:
`wscene2 window adjust <videowall>-<wscene> <wname> <X> <Y> <H> <W> [success|failure]`

Command Example: Reposition and resize a window within a scene
```
wscene2 window adjust OfficeVW-windowscene window1 1 1 2 2
```

Response Example: Command acknowledgment
```
wscene2 window adjust OfficeVW-windowscene window1 1 1 2 2 success
```

`videowall` = Preconfigured video wall name

`wscene` = Preconfigured Scene name

`wname` = Unique window name

`X` = Window's top left decoder horizontal position in display array

`Y` = Window's top left decoder vertical position in display array

`H` = Window's horizontal size in number of displays

`W` = Window's vertical size in number of displays

Command applies to:

NHD-110-RX, NHD-210-RX

NHD-400-RX, NHD-500-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Command is effective immediately and does not save to the default Scene.
Horizontal and vertical positions are relative to the top left display.
Window names must use names of windows currently open in the Scene.

## Change the layer for an open window within a 'video wall within a wall' Scene

Command structure:
`wscene2 window move <videowall>-<wscene> <wname> <layer>`

Response structure:
`wscene2 window move <videowall>-<wscene> <wname> <layer> [success|failure]`

Command Example: Move window to top layer
```
wscene2 window move OfficeVW-windowscene window1 top
```

Response Example: Command acknowledgment
```
wscene2 window move OfficeVW-windowscene window1 top success
```

`videowall` = Preconfigured video wall name

`wscene` = Preconfigured Scene name

`wname` = Unique window name

`layer` = Window's layer reference
      `up`
      `down`
      `top`
      `bottom`

Command applies to:

NHD-110-RX, NHD-210-RX

NHD-400-RX, NHD-500-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Command is effective immediately and does not save to the default Scene.
Window names must use names of windows currently open in the Scene.

# 11. Multiview

## 11.1 Multiview Decoders – Single Encoder

Use this command to apply a single source to a display. Where available at the NetworkHD endpoints, equivalent commands to stream matrix switching can be achieved on Multiview decoders.

| Assign a single TX audio and video stream to a Multiview RX | |
|---|---|
| Command structure:<br>`mview set <RX> ([tile|overlay]) <TX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure:<br>`mview set <RX> ([tile|overlay]) <TX> [success|failure]` | **RX** = Decoder reference – Alias or hostname |
| Command Example: Assign a TX to an RX<br>`mview set display1 source1` | `tile` = Use for non-overlapping tile layouts |
| Response Example: Command acknowledgment<br>`mview set display1 source1 success` | `overlay` = Use for overlapping tile layouts |
| Command applies to:<br>NHD-220-RX, NHD-250-RX | |

Notes:
Multiview only decoders do not respond to `matrix set` commands found in section 6 – Media Stream Matrix Switch Commands, equivalent functions can be applied using this command.
The use of `tile` or `overlay` in the command is optional and can be used where switching to/from Preset or Custom Tile layouts where a small switching delay advantage can be obtained when staying in the same mode.
For NetworkHD 600, use `matrix set` commands found in section 6 – Media Stream Matrix Switch Commands.

## 11.2 Multiview Decoders – Preset Tile Layouts

Use this command to apply preset Multiview layouts to a decoder. Where available at the NetworkHD endpoints, Multiview layouts configured in the NetworkHD Console software utility can be recalled and applied to NetworkHD decoders.

| Apply a preset Multiview layout to an RX | |
|---|---|
| Command structure:<br>`mscene active <RX> <lname>` | **RX** = Decoder reference – Alias or hostname |
| Response structure:<br>`mscene active <RX> <lname> [success|failure]` | `lname` = Preconfigured layout name |
| Command Example: Apply a layout to a decoder<br>`mscene active display1 gridlayout` | |
| Response Example: Command acknowledgment<br>`mscene active display1 gridlayout success` | |
| Command applies to:<br>NHD-220-RX, NHD-250-RX<br>NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Multiview layout names are given to preset layouts when they are created. These are listed and selected in NetworkHD Console.
When Multiview layouts are configured in Console, the layout tiles will have default encoders assigned to them.

## Change the TX for a tile within a preset Multiview layout

| | |
|---|---|
| Command structure:<br>`mscene change <RX> <lname> <tile> <TX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure:<br>`mscene change <RX> <lname> <tile> <TX> [success\|failure]` | **RX** = Decoder reference – Alias or hostname |
| Command Example: Reposition and resize a window within a scene | **lname** = Preconfigured layout name |
| `mscene change display1 gridlayout 1 source1` | **tile** = Tile reference |
| Response Example: Command acknowledgment | |
| `mscene change display1 gridlayout 1 source1 success` | |

Command applies to:

NHD-220-RX, NHD-250-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:

Command is effective immediately and does not save to the default Scene.

Tile reference is the integer reference inferred to the target tile when the layout is created. This value can be referenced when configuring the layout in NetworkHD Console.

## Change the audio mode within a preset Multiview layout

| | |
|---|---|
| Command structure:<br>`mscene set audio <RX> <lname> [window\|separate] <tile\|TX>` | **TX** = Encoder reference – Alias or hostname |
| | **RX** = Decoder reference – Alias or hostname |
| Response structure:<br>`mscene set audio <RX> <lname> [window\|separate] <tile\|TX> [success\|failure]` | **lname** = Preconfigured layout name |
| | **tile** = Tile reference |
| Command Example: Apply RX audio to a specific TX | |
| `mscene set audio display1 gridlayout separate source1` | **window** = RX audio follows video for Tile |
| Response Example: Command acknowledgment | |
| `mscene set audio display1 gridlayout separate source1 success` | **separate** = RX audio is assigned to TX directly |
| Command Example: Apply RX audio to a specific tile | |
| `mscene set audio display1 gridlayout window 1` | |
| Response Example: Command acknowledgment | |
| `mscene set audio display1 gridlayout window 1 success` | |

Command applies to:

NHD-220-RX, NHD-250-RX

Notes:

Command saves audio mode to RX and is not effective immediately. Use the `mscene active` command to apply the audio mode.

Tile reference is the integer reference inferred to the target tile when the layout is created. This value can be referenced when configuring the layout in NetworkHD Console.

For NetworkHD 600 Series, use the relevant audio commands found in section 6.3 – Stream Matrix Switching – Audio Stream Breakaway.

## 11.3 Multiview Decoders – Custom Tile Layouts

Use this command to apply a custom Multiview layout to a decoder. Where available at the NetworkHD endpoints, custom Multiview layouts can be created and applied to NetworkHD decoders.

| Apply a custom Multiview layout to an RX | |
|---|---|
| Command structure:<br>`mview set <RX> [tile|overlay]`<br>`<TX1>:<X1>_Y1>_<W1>_<H1>:[fit|stretch]`<br>`<TX2>:<X2>_Y2>_<W2>_<H2>:[fit|stretch] …`<br>`<TXn>:<Xn>_Yn>_<Wn>_<Hn>:[fit|stretch]` | `TX` = Encoder reference – Alias or hostname |
| | `RX` = Decoder reference – Alias or hostname |
| Response structure:<br>`mview set <RX> [tile|overlay]`<br>`<TX1>:<X1>_Y1>_<W1>_<H1>:[fit|stretch]`<br>`<TX2>:<X2>_Y2>_<W2>_<H2>:[fit|stretch] …`<br>`<TXn>:<Xn>_Yn>_<Wn>_<Hn>:[fit|stretch] [success|failure]` | `Xn` = Tile's top left pixel – horizontal reference |
| | `Yn` = Tile's top left pixel – vertical reference |
| | `Wn` = Tile horizontal size in pixels |
| | `Wn` = Tile vertical size in pixels |
| Command Example: Apply a 2x2 tile-mode layout to an RX<br>`mview set display1 tile source1:0_0_960_540:fit`<br>`source2:960_0_960_540:fit`<br>`source3:0_540_960_540:fit`<br>`source4:960_540_960_540:fit` | `tile` = Use for non-overlapping tile layouts |
| | `overlay` = Use for overlapping tile layouts |
| Response Example: Command acknowledgment<br>`mview set display1 tile source1:0_0_960_540:fit`<br>`source2:960_0_960_540:fit`<br>`source3:0_540_960_540:fit`<br>`source4:960_540_960_540:fit success` | `fit` = Scales source to fit window without distortion |
| | `stretch` = Scales source to fit window completely |
| Command Example: Apply a PiP overlay-mode layout to an RX<br>`mview set display1 overlay source1:100_50_256_144:fit`<br>`source2:0_0_1920_1080:fit` | |
| Response Example: Command acknowledgment<br>`mview set display1 overlay source1:100_50_256_144:fit`<br>`source2:0_0_1920_1080:fit success` | |
| Command applies to: | |
| NHD-220-RX, NHD-250-RX | |

Notes:
Command is effective immediately but does not save to NHD-CTL.
Tile horizontal and vertical start pixels are relative to the top left canvas pixel but can be no greater that the bottom right canvas pixel.
Tile size must ensure that the tile will not exceed the available pixels in the canvas.
`overlay` mode is only supported on NetworkHD decoders that support this feature.
`fit` scaling will fill the tile with the source video and not distort the source video, but it may result in some missing video if the aspect ratio is not matched to the tile.
`stretch` scaling will fill the tile with the source video, but it may introduce distortion if the aspect ratio is not matched to the tile.
The tile layer order follows the order in which tiles are defined. The topmost tile is the first to be defined.

## Change the audio source within a custom Multiview layout

| | |
|---|---|
| Command structure:<br>`mview set audio <RX> separate <TX>` | **TX** = Encoder reference – Alias or hostname |
| Response structure:<br>`mview set audio <RX> separate <TX> [success\|failure]` | **RX** = Decoder reference – Alias or hostname |
| Command Example: Apply RX audio to a specific TX<br>`mview set audio display1 separate source1` | |
| Response Example: Command acknowledgment<br>`mview set audio display1 separate source1 success` | |

Command applies to:
NHD-220-RX, NHD-250-RX

Notes:
Command is effective immediately but does not save to NHD-CTL.
Default behaviour for an RX using a custom Multiview layout is to have no audio.

# 12.  API Notifications

## 12.1 Enable / Disable Notifications

Use this command to enable or disable API notifications. Where available, some API notifications can be suppressed to remove large amounts of data being sent to the 3rd-party control system.

| Enable or disable the CEC API notifications | |
|---|---|
| Command structure:<br>`config set device cec notify [on\|off] <TX\|RX\|bulk>` | **TX** = Encoder reference – Alias or hostname |
| Response structure (command mirror):<br>`config set device cec notify [on\|off] <TX\|RX\|bulk>` | **RX** = Decoder reference – Alias or hostname |
| Command Example: Turn on CEC notifications for all RX<br>`config set device cec notify on ALL_RX` | **bulk** = multiple devices<br>    **ALL_DEV** = all endpoints<br>    **ALL_TX** = all encoders<br>    **ALL_RX** = all decoders |
| Response Example: Command acknowledgment<br>`config set device cec notify on ALL_RX` | |
| Command applies to:<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |

North America: 844.280.WYRE (9973) | UK/EMEA: +44 (0) 1793 230 343
Email: Support@WyreStorm.com

## 12.2 Endpoint Notifications

Use this command for notifications that originate from an endpoint. Where available, endpoints will use API notifications to signal a change in status or availability of data.

| Endpoint online status notification | |
| --- | --- |
| Notification structure:<br>`notify endpoint [+|-] <TX|RX>` | **TX** = Encoder reference – Alias or hostname<br><br>**RX** = Decoder reference – Alias or hostname |
| Notification example: RX has gone offline<br>`notify endpoint – display1` | |
| Notification example 2: TX has come online<br>`notify endpoint + source1` | |
| Command applies to:<br>NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

| Endpoint CEC data notification | |
| --- | --- |
| Notification structure:<br>`notify cecinfo <TX|RX> "<cecdata>"` | **TX** = Encoder reference – Alias or hostname<br><br>**RX** = Decoder reference – Alias or hostname<br><br>`cecdata` = CEC data string |
| Notification example: RX has received CEC data<br>`notify cecinfo display1 "FF36"` | |
| Command applies to:<br>NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX | |

| Endpoint Infrared data notification | |
| --- | --- |
| Notification structure:<br>`notify irinfo <TX|RX> "<irdata>"` | **TX** = Encoder reference – Alias or hostname<br><br>**RX** = Decoder reference – Alias or hostname<br><br>`irdata` = Infrared data string |
| Notification example: RX has received infrared data<br>`notify irinfo display1 "0000 0067 0000 0015 0060 0018 0030 0018`<br>`0018 0018 0030 0018 0018 0018 0030 0018 0018 0018 0018 0018`<br>`0018 0030 0018 0018 0018 0030 0018 0030 0018 0018 0018 0030 0018`<br>`0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE"` | |
| Command applies to:<br>NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX | |

Notes:
Infrared data will be received if the correct setting is applied to endpoints using the `matrix infrared2 set` command found in section 6.5 - Stream Matrix Switching – Infrared Stream Breakaway.

## Endpoint RS-232 data notification

Notification structure:
**notify serialinfo <TX|RX> [hex|ascii] <infolen>:<CR><LF><sdata>**

Notification example 1: RX has received RS-232 data in Hex format
```
notify serialinfo display1 hex 371:
4c 6f 72 65 6d 20 69 70 73 75 6d 20 64 6f 6c 6f 72 20 73 69 74
20 61 6d 65 74 2c 20 63 6f 6e 73 65 63 74 65 74 75 72 20 61 64
69 70 69 73 63 69 6e 67 20 65 6c 69 74 2c 20 73 65 64 20 64 6f
20 65 69 75 73 6d 6f 64 20 74 65 6d 70 6f 72 20 69 6e 63 69 64
69 64 75 6e 74 20 75 74 20 6c 61 62 6f 72 65 20 65 74 20 64 6f
6c 6f 72 65 20 6d 61 67 6e 61 20 61 6c 69 71 75 61 0D 0A
```

Notification example 1: RX has received RS-232 data in ASCII format
```
notify serialinfo display1 ascii 122:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**infolen** = length of formatted data string (sdata) in characters:
   for ASCII: infolen = number of received bytes

   <u>for Hex</u>: (infolen + 1)/3 = number of received bytes

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

**sdata** = RS-232 formatted data string

Command applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX
NHD-400-TX (IW only), NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
RS-232 data will generate a notification automatically on NetworkHD 100, 200 and 400 Series endpoints.
On NetworkHD 600 Series endpoints, data will be received if the correct setting is applied to endpoints using the `matrix serial2 set` command found in section 6.6 - Stream Matrix Switching – RS-232 Stream Breakaway.

## Endpoint video input status notification

Notification structure:
**notify video [lost|found] <TX|RX> (<TX>)**

Notification example: RX has (re)established connection with TX
```
notify video found display1 source1
```

Notification example: TX has lost source input video
```
notify video lost source1
```

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**lost** = video input has been lost from endpoint

**found** = video input has been (re)established with endpoint

Command applies to:
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

Notes:
Video input at a TX means the video signal entering the video input port. This is not the same thing as hotplug detection, e.g. the source has entered a standby mode and is not outputting video but is still physically connected.
Video input at an RX means the AV over IP video stream from an encoder. It is not concerned with video output connections.
The field in parenthesis is only used when an RX reports a (re)established connection to a TX and informs of the connected TX as the last variable.

# 13. API Query Commands

## 13.1 Query Commands – System Configuration

Use this command to query NetworkHD system configuration parameters.

.

| Query NHD-CTL version | |
|---|---|
| Command structure:<br>`config get version` | **api** = API version number |
| Response structure:<br>`API version: v<api><CR><LF>System version: v<web>(v<core>)` | **web** = Webserver version number |
| | **core** = System core version number |
| Command Example: Get NHD-CTL version<br>`config get version` | **CR** = Carriage Return byte (0x0D) (CR) (\r) |
| Response Example: Return data<br>`API version: v1.21`<br>`System version: v8.3.1(v8.3.8)` | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Command applies to:<br>NHD-000-CTL, NHD-CTL-PRO | |

| Query NHD-CTL "AV" port network interface details | |
|---|---|
| Command structure:<br>`config get ipsetting` | **ipv4** = IPv4 address value |
| Response structure:<br>`ipsetting is: ip4addr <ipv4> netmask <nm> gateway <gw>` | **nm** = Netmask value |
| | **gw** = IPv4 default gateway address value |
| Command Example: Get NHD-CTL "AV" port network interface details<br>`config get ipsetting` | |
| Response Example: Return data<br>`ipsetting is: ip4addr 169.254.1.1 netmask 255.255.0.0 gateway`<br>`169.254.1.254` | |
| Command applies to:<br>NHD-000-CTL, NHD-CTL-PRO | |

Note:
Values can be assigned using the NetworkHD Console application.
Default gateway values will be listed but might not be active for the interface. The NHD-CTL can only have one active gateway IP. The interface with the disabled gateway will list gw as 0.0.0.0

## Query NHD-CTL "CONTROL" port network interface details

| | |
|---|---|
| Command structure:<br>**config get ipsetting2** | **ipv4** = IPv4 address value |
| Response structure:<br>**ipsetting2 is: ip4addr <ipv4> netmask <nm> gateway <gw>** | **nm** = Netmask value<br><br>**gw** = IPv4 default gateway address value |
| Command Example: Get NHD-CTL "CONTROL" port network interfacedetails | |

```
config get ipsetting2
```

Response Example: Return data

```
ipsetting2 is: ip4addr 192.168.11.243 netmask 255.255.255.0
gateway 192.168.11.1
```

Command applies to:
NHD-000-CTL, NHD-CTL-PRO

Note:
Values can be assigned using the NetworkHD Console application.
Default gateway values will be listed but might not be active for the interface. The NHD-CTL can only have one active gateway IP. The interface with the disabled gateway will list gw as 0.0.0.0

## Query online endpoint names

| | |
|---|---|
| Command structure:<br>**config get devicelist** | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>**devicelist is <TX1\|RX1> <TX2\|RX2> … <TXn\|RXn>** | **RXn** = Decoder reference – Alias or hostname |
| Command Example: Get online endpoints | |

```
config get devicelist
```

Response Example: Return data

```
devicelist is source1 source2 display1 display2 display3
```

Command applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Note:
This will not return offline device names.

## Query general information for all TX or RX saved in the NHD-CTL

| | |
|---|---|
| Command structure:<br>**config get devicejsonstring** | **JSONdata** = data in JSON format |
| Response structure:<br>**device json string: <JSONdata>** | |
| Command Example: Get online endpoints | |

```
config get devicejsonstring
```

Response Example: See Appendix A

Command applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Note:
Refer to Appendix A for further JSON format response data information.

## 13.2 Query Commands – Device Configuration

Use this command to query NetworkHD endpoint configurations. Where available at the NetworkHD endpoints, endpoints can be queried to return current status information.

| Query endpoint Alias and Hostname | |
|---|---|
| Command structure:<br>`config get name (<alias\|hostname>)` | `alias` = endpoint Alias<br><br>`hostname` = endpoint Hostname |
| Response structure:<br>`<hostname>'s alias is <alias\|null>` | |
| Command Example 1: Get all endpoint Alias and Hostname<br>`config get name` | |
| Response Example:<br>`NHD-400-TX-E4CE02104E55's alias is source1`<br>`NHD-400-TX-E4CE02104E56's alias is source2`<br>`NHD-400-RX-E4CE02104A57's alias is display1`<br>`NHD-400-RX-E4CE02104A58's alias is null` | |
| Command Example 2: Get endpoint Alias and Hostname<br>`config get name NHD-400-RX-E4CE02104A57` | |
| Response Example:<br>`NHD-400-RX's alias is display1` | |

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Note:
Alias may be null if the endpoint has not had the default Alias changed using the NetworkHD Console utility.

| Query TX or RX device working parameters | |
|---|---|
| Command structure:<br>`config get device info (<TX\|RX>)` | `TXn` = Encoder reference – Alias or hostname<br><br>`RXn` = Decoder reference – Alias or hostname<br><br>`JSONdata` = data in JSON format |
| Response structure:<br>`devices json info: <JSONdata>` | |
| Command Example 1: Get all TX and RX device working parameters<br>`config get device info` | |
| Response Example: See [Appendix A](Appendix A) | |
| Command Example 2: Get TX device working parameters<br>`config get device info source1` | |
| Response Example: See [Appendix A](Appendix A) | |

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Note:
Refer to [Appendix A](Appendix A) for further JSON format response data information.

## Query TX or RX device real-time status

| | |
|---|---|
| Command structure:<br>`config get device status (<TX\|RX>)` | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>`devices status info: <JSONdata>` | **RXn** = Decoder reference – Alias or hostname |
| Command Example 1: Get all TX and RX device real-time status | **JSONdata** = data in JSON format |
| `config get device status` | |
| Response Example: See Appendix A | |
| Command Example 2: Get TX device real-time status | |
| `config get device status source1` | |
| Response Example: See Appendix A | |

Command applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Note:
Refer to Appendix A for further JSON format response data information.

## 13.3 Query Commands – Stream Matrix Switching

Use this command to query stream matrix assignments. Where available at the NetworkHD endpoints, endpoints can be queried to return the current stream matrix assignments between endpoints.

| Query RX assignments where using all primary media streams | |
|---|---|
| Command structure:<br>`matrix get (<RX1>) (<RX2>) … (<RXn>)` | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>`matrix information:<CR><LF><TXn\|NULL> <RX1><CR><LF><TXn\|NULL> <RX2> … <CR><LF><TXn\|NULL> <RXn>` | **RXn** = Decoder reference – Alias or hostname<br><br>**CR** = Carriage Return byte (0x0D) (CR) (\r) |
| Command Example: Get RX assignments where using all primary media streams<br>`matrix get` | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Response Example: Return data<br>`matrix information:`<br>`Source1 Display1`<br>`Source1 Display2`<br>`Source2 Display3`<br>`NULL Display4` | |

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set` command found in section 6.1 – Stream Matrix Switching – All Media
3) The RX has been assigned discrete media streams. E.g. using the `matrix video set`, `matrix audio set` commands found in section 6 – Media Stream Matrix Switch Commands

The response does not return assignments for RXs using Multiview.

## Query RX stream assignments for video output

Command structure:
```
matrix video get (<RX1>) (<RX2>) … (<RXn>)
```

Response structure:
```
matrix video information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for primary video streams
```
matrix video get
```

Response Example: Return data
```
matrix video information:
Source1 Display1
Source1 Display2
Source2 Display3
NULL Display4
```

Command Example 2: Get RX assignments for primary video streams
```
matrix video get display1 display3
```

Response Example: Return data
```
matrix video information:
Source1 Display1
Source2 Display3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix video set` command found in section 6 – Media Stream Matrix Switch Commands
The response does not return assignments for RXs using Multiview.

## Query RX stream assignments for HDMI audio output

Command structure:
```
matrix audio get (<RX1>) (<RX2>) … (<RXn>)
```

Response structure:
```
matrix audio information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for HDMI audio streams
```
matrix audio get
```

Response Example: Return data
```
matrix audio information:
Source1 Display1
Source1 Display2
Source2 Display3
NULL Display4
```

Command Example 2: Get RX assignments for HDMI audio streams
```
matrix audio get display1 display3
```

Response Example: Return data
```
matrix audio information:
Source1 Display1
Source2 Display3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix audio set` command found in section 6 – Media Stream Matrix Switch Commands

The response does not return assignments for RXs using Multiview.

## Query RX stream assignments for analog audio output

Command structure:
```
matrix audio2 get (<RX1>) (<RX2>) … (<RXn>)
```

Response structure:
```
matrix audio2 information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for analog audio streams
```
matrix audio2 get
```

Response Example: Return data
```
matrix audio2 information:
Source1 Display1
Source1 Display2
Source2 Display3
NULL Display4
```

Command Example 2: Get RX assignments for analog audio streams
```
matrix audio2 get display1 display3
```

Response Example: Return data
```
matrix audio2 information:
Source1 Display1
Source2 Display3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix audio2 set` command found in section 6 – Media Stream Matrix Switch Commands
Note that while an assignment may be made to a TX, the audio path may not be set in the RX. See the `config set device audio2source` command found in section 7.2 – Port Switching - Audio
The response does not return assignments for RXs using Multiview.

## Query RX stream assignments for ARC

Command structure:
```
matrix audio3 get (<RX1>) (<TX1>)
```

Response structure:
```
matrix audio3 information:<CR><LF>
<RX1><CR><LF><TX1>
```

Command Example 1: Get all RX assignments for ARC streams
```
matrix audio3 get
```

Response Example: Return data
```
matrix audio3 information:
Display1 Source1
Display2 Source3
Display5 Source2
```

Command Example 2: Get specific RX assignment for ARC streams
```
matrix audio3 Display1
```

Response Example: Return data
```
matrix audio3 information:
Display1 Source3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:

NHD-500-TX, NHD-500-RX

## Query RX stream assignments for USB

Command structure:
```
matrix usb get (<RX1>) (<RX2>) … (<RXn>)
```

Response structure:
```
matrix usb information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for USB
```
matrix usb get
```

Response Example: Return data
```
matrix usb information:
Source1 Display1
Source1 Display2
Source2 Display3
NULL Display4
```

Command Example 2: Get RX assignments for USB
```
matrix usb get display1 display3
```

Response Example: Return data
```
matrix usb information:
Source1 Display1
Source2 Display3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:

NHD-110-TX, NHD-110-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix usb set` command found in section 6 – Media Stream Matrix Switch Commands

## Query RX stream assignments for infrared

**Command structure:**
```
matrix infrared get (<RX1>) (<RX2>) … (<RXn>)
```

**Response structure:**
```
matrix infrared information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for USB
```
matrix infrared get
```

Response Example: Return data
```
matrix infrared information:
Source1 Display1
Source1 Display2
Source2 Display3
NULL Display4
```

Command Example 2: Get RX assignments for USB
```
matrix infrared get display1 display3
```

Response Example: Return data
```
matrix infrared information:
Source1 Display1
Source2 Display3
```

**Command applies to:**

NHD-110-TX, NHD-110-RX

NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix infrared set` command found in section 6 – Media Stream Matrix Switch Commands

**Query TX or RX stream assignments for infrared receiver port**

Command structure:
```
matrix infrared2 get (<TX1|RX1>) (<TX2|RX2>) … (<TXn|RXn>)
```

Response structure:
```
matrix infrared2 information:<CR><LF><TX1|RX1> <mode>
(<TXn|RXn>)<CR><LF><TX2|RX2> <mode> (<TXn|RXn>) …
<CR><LF><TXn|RXn> <mode> (<TXn|RXn>)
```

Command Example 1: Get all endpoint IR receiver assignments
```
matrix infrared2 get
```

Response Example: Return data
```
matrix infrared2 information:
source1 single display1
display1 api
source2 api
display2 null
```

Command Example 2: Get endpoint IR assignments
```
matrix infrared2 get display1 source1
```

Response Example: Return data
```
matrix infrared2 information:
display1 api
source1 single display1
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

**mode** = Operational mode:
    **single** = Assigned to endpoint
    **api** = Assigned to API notification
    **all** = Assigned to all other endpoints
    **null** = Assigned to no endpoints

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The endpoint IR receiver assignment will show NULL if:
1) The endpoint is in a default state and has not yet been assigned a TX
2) The endpoint IR receiver has been assigned NULL using the `matrix set`, `matrix infrared2 set` command found in section 6 – Media Stream Matrix Switch Commands

## Query RX stream assignments for RS-232

Command structure:
```
matrix serial get (<RX1>) (<RX2>) … (<RXn>)
```

Response structure:
```
matrix serial information:<CR><LF><TXn|NULL>
<RX1><CR><LF><TXn|NULL> <RX2> … <CR><LF><TXn|NULL> <RXn>
```

Command Example 1: Get all RX assignments for RS-232
```
matrix serial get
```

Response Example: Return data
```
matrix serial information:
Source1 Display1
Source1 Display2
Source2 Display3
null Display4
```

Command Example 2: Get RX assignments for RS-232
```
matrix serial get display1 display3
```

Response Example: Return data
```
matrix serial information:
Source1 Display1
Source2 Display3
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

Command applies to:
NHD-400-TX, NHD-400-RX, NHD-500-TX, NHD-500-RX

Notes:
The RX assignment will show NULL if:
1) The RX is in a default state and has not yet been assigned a TX
2) The RX has been assigned NULL using the `matrix set`, `matrix serial set` command found in section 6 – Media Stream Matrix Switch Commands

## Query TX or RX stream assignments for RS-232 port

Command structure:
```
matrix serial2 get (<TX1|RX1>) (<TX2|RX2>) … (<TXn|RXn>)
```

Response structure:
```
matrix serial2 information:<CR><LF><TX1|RX1> <mode>
(<TXn|RXn>)<CR><LF><TX2|RX2> <mode> (<TXn|RXn>) …
<CR><LF><TXn|RXn> <mode> (<TXn|RXn>)
```

Command Example 1: Get all endpoint RS-232 receiver assignments
```
matrix serial2 get
```

Response Example: Return data
```
matrix serial2 information:
source1 single display1
display1 api
source2 api
display2 null
```

Command Example 2: Get endpoint RS-232assignments
```
matrix serial2 get display1 source1
```

Response Example: Return data
```
matrix serial2 information:
display1 api
source1 single display1
```

**TXn** = Encoder reference – Alias or hostname

**RXn** = Decoder reference – Alias or hostname

**CR** = Carriage Return byte (0x0D) (CR) (\r)

**LF** = Line Feed byte (0x0A) (LF) (\n)

**mode** = Operational mode:
- **single** = Assigned to endpoint
- **api** = Assigned to API notification
- **all** = Assigned to all other endpoints
- **null** = Assigned to no endpoints

Command applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

Notes:
The endpoint RS-232 port assignment will show NULL if:
1) The endpoint is in a default state and has not yet been assigned a TX
2) The endpoint IR receiver has been assigned NULL using the `matrix set`, `matrix serial2 set` command found in section 6 – Media Stream Matrix Switch Commands

# 13.4 Query Commands – Video Walls

Use this command to query elements of Video wall configurations. Where available at the NetworkHD endpoints, endpoints can be queried to return the current configurations related to video walls.

| Query 'standard video wall' Scene list | |
|---|---|
| Command structure:<br>**scene get** | **videowall** = Preconfigured video wall name |
| Response structure:<br>**scene list:\<CR>\<LF>\<videowall1>-\<scene1> \<videowall1>-\<scene2> … \<videowalln>-\<scenen>** | **scene** = Preconfigured Scene name<br><br>**CR** = Carriage Return byte (0x0D) (CR) (\r) |
| Command Example: Get list of all Scenes | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| **scene get** | |
| Response Example: Return data | |
| **scene list:**<br>**OfficeVW-Splitmode OfficeVW-Combined** | |
| Command applies to: | |
| NHD-110-RX, NHD-210-RX | |
| NHD-400-RX, NHD-500-RX | |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Video wall names are given to physical video wall arrays when created in the NetworkHD Console software utility.
Scene names are given to Scenes when created in the NetworkHD Console software utility.

| Query 'standard video wall' logical screen list | |
|---|---|
| Command structure:<br>**vw get** | **TXn** = Encoder reference – Alias or hostname |
| Response structure:<br>**Video wall information:\<CR>\<LF>\<videowall1>-\<scene1>_\<Lscreen1> \<TX>\<CR>\<LF>Row 1: \<RX1> \<RX2>\<Row 2: \<RX3> \<RX4> … \<CR>\<LF>\<videowalln>-\<scenen>_\<Lscreenn> \<TX>\<CR>\<LF>Row 1: \<RXn> \<RXn>Row 2: \<RXn> \<RXn> …** | **videowall** = Preconfigured video wall name<br><br>**scene** = Preconfigured Scene name<br><br>**Lscreen** = Preconfigured logical screen name |
| Command Example: Get list of all logical screens | **CR** = Carriage Return byte (0x0D) (CR) (\r) |
| **vw get** | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Response Example: Return data | |
| **Video wall information:**<br>**OfficeVW-Combined_TopTwo source1**<br>**Row 1: display1 display2**<br>**OfficeVW-AllCombined_AllDisplays source2**<br>**Row 1: display1 display2 display3**<br>**Row 2: display4 display5 display6**<br>**Row 3: display7 display8 display9** | |
| Command applies to: | |
| NHD-110-RX, NHD-210-RX | |
| NHD-400-RX, NHD-500-RX | |
| NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX | |

Notes:
Video wall names are given to physical video wall arrays when created in the NetworkHD Console software utility.
Scene names are given to Scenes when created in the NetworkHD Console software utility.
Logical screen names are given to logical screens when created in the NetworkHD Console software utility.

Command structure:
**config set device osd param tx-1 position_x position_y FFFFFFFF 1 hostname1**

**videowall** = Preconfigured video wall name

Response structure:
**wscene2 list:<CR><LF><videowall1>-<wscene1> <videowall1>-<wscene2> … <videowalln>-<wscenen>**

**wscene** = Preconfigured Scene name

**CR** = Carriage Return byte (0x0D) (CR) (\r)

Command Example: Get list of all Scenes
**wscene2 get**

**LF** = Line Feed byte (0x0A) (LF) (\n)

Response Example: Return data
**wscene2 list:**
**OfficeVW-windowscene1 OfficeVW-windowscene2**

Command applies to:

NHD-110-RX, NHD-210-RX

NHD-400-RX, NHD-500-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Video wall names are given to physical video wall arrays when created in the NetworkHD Console software utility.
Scene names are given to Scenes when created in the NetworkHD Console software utility.

## 13.5 Query Commands – Multiview

Use this command to query elements of Multiview configurations. Where available at the NetworkHD endpoints, endpoints can be queried to return the current configurations related to Multiview.

| Query preset Multiview layout list | |
|---|---|
| Command structure:<br>`mscene get (<RX>)` | **RX** = Decoder reference – Alias or hostname |
| Response structure:<br>`mscene list:<CR><LF><RX> <lname1> <lname2> …`<br>`<lnamen><CR>LF><RXn> <lname3> <lname4> …` | **lname** = Preconfigured layout name |
| Command Example 1: Get list of all layouts | **CR** = Carriage Return byte (0x0D) (CR) (\r) |
| `mscene get` | **LF** = Line Feed byte (0x0A) (LF) (\n) |
| Response Example: Return data | |
| `mscene list:`<br>`display5 gridlayout piplayout`<br>`display6 pip2layout`<br>`display7 grid5layout grid6layout` | |
| Command Example 2: Get list of layouts for RX | |
| `mscene get display6` | |
| Response Example: Return data | |
| `mscene list:`<br>`display6 pip2layout` | |

Command applies to:

NHD-220-RX, NHD-250-RX

NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

Notes:
Multiview layout names are given to preset layouts when they are created. These are listed and selected in NetworkHD Console.

## Query custom Multiview layout list

Command structure:
```
mview get (<RX>)
```

Response structure:
```
mview information:<CR><LF>
<RX1> [tile|overlay] <TX1>:<X1>_Y1>_<W1>_<H1>:[fit|stretch]
<TX2>:<X2>_Y2>_<W2>_<H2>:[fit|stretch] …
<TXn>:<Xn>_Yn>_<Wn>_<Hn>:[fit|stretch]<CR><LF>
<RXn> [tile|overlay] <TX1>:<X1>_Y1>_<W1>_<H1>:[fit|stretch]
<TX2>:<X2>_Y2>_<W2>_<H2>:[fit|stretch] …
<TXn>:<Xn>_Yn>_<Wn>_<Hn>:[fit|stretch] …
```

Command Example 1: Get list of all layouts
```
mview get
```

Response Example: Return data
```
mview information:
display10 tile source1:0_0_960_540:fit source2:960_0_960_540:fit
source3:0_540_960_540:fit
source4:960_540_960_540:fit
display11 overlay source1:100_50_256_144:fit
source2:0_0_1920_1080:fit
```

Command Example 2: Get list of layouts for RX
```
mview get display11
```

Response Example: Return data
```
mview information:
display11 overlay source1:100_50_256_144:fit
source2:0_0_1920_1080:fit
```

Command applies to:
NHD-220-RX, NHD-250-RX

---

**TX** = Encoder reference – Alias or hostname

**RX** = Decoder reference – Alias or hostname

**Xn** = Tile's top left pixel – horizontal reference

**Yn** = Tile's top left pixel – vertical reference

**Wn** = Tile horizontal size in pixels

**Wn** = Tile vertical size in pixels

**tile** = Use for non-overlapping tile layouts

**overlay** = Use for overlapping tile layouts

**fit** = Scales source to fit window without distortion

**stretch** = Scales source to fit window completely

# 14.  Video Preview Streams

Use these commands to obtain the low-bandwidth video preview stream from a NetworkHD encoder. Where available at the NetworkHD encoders, low-bandwidth, unicast preview stream video can be rendered by a capable 3rd-party application.

The communication topology for preview streams differs from the preceding chapters. Unlike the general control API, preview streams are obtained directly from each NetworkHD encoder using HTTP.

## 14.1 IPv4 Network Communications

Each NetworkHD encoder has 1 network interface. Video preview stream communications are achieved by connecting with the network interface port to each encoder directly.

When using factory default settings, the NetworkHD encoder network interfaces have the following parameters:

| NetworkHD Series | Mode | IPv4 Address | Subnet Mask | Default Gateway IPv4 Address |
|---|---|---|---|---|
| 100 / 400 / 500 | DHCP/AutoIP | (AutoIP Range) 169.254.xxx.xxx | 255.255.0.0 | 0.0.0.0 (none) |
| 600 | 600 series encoders do not support 3rd-party access to preview streams. | | | |

To obtain actual IP addresses for encoders in a configured NetworkHD system, use the NetworkHD Console Utility or an API query command, such as **config get device info** found in section 13.1 - Query Commands – Device Configuration.

Communications with each encoder should use the HTTP (RFC 2616) protocol using the GET method on TCP port 80. Each encoder is a web server.

## 14.2 Video Preview Stream Format

Video preview streams use the Motion JPEG (MJPEG) codec. The video format is Planar 4:2:0 YUV Full Scale. The color space is ITU-R BT601.

## 14.3 Preview Streams API Reference

| URL structure for a HTTP GET request | |
|---|---|
| URL structure:<br>**http://<IPv4>/stream(?resolution=<res>&fps=<fps>&bitrate=<br>)** | **IPv4** = Encoder IPv4 address<br><br>**res** = Video resolution<br>    CIF<br>    480P<br>    720x480<br>    1280x720<br><br>**fps** = Video resolution<br>    **1** to **30** (integer)<br><br>**br** = Target bitrate<br>    **128** to **2000** (kb/s, integer) |
| URL Example 1: Get preview stream (using default or last known parameters)<br>**http://169.254.6.45/stream** | |
| URL Example 2: Get preview stream using defined parameters<br>**http://169.254.6.45/stream?resolution=480P&fps=5&bitrate=500** | |

Command applies to:

NHD-110-TX, NHD-100-TX, NHD-140-TX, NHD-200-TX

NHD-400-TX, NHD-500-TX

Notes:

NetworkHD 400 & 500 Series encoders do not support stream parameters and will ignore them

If HTTPS is enabled on encoders, the URL will need to use 'https://' rather than 'http://'

# 15.  Video Stream Text Overlay

| Configure Text Overlay | |
|---|---|
| Command structure:<br>**config set device osd param <text> <position_x> <position_y> <text_color> <text_size> <TX>** | **text** = The text to be displayed<br><br>**position_x** = X coordinate of text; must be value between 0~1920 |
| Response structure:<br>**config set device osd param <text> <position_x> <position_y> <text_color> <text_size> <TX>** | **position_y** = Y coordinate of text; must be value between 0~1080 |
| Command Example: Get list of all Scenes<br>**config set device osd param Hello 0 0 FFFFFFFF 1 Input1** | **text_color** = Color of text in hex format, examples below: |
| Response Example: Return data<br>**receive: config set device osd param Hello 0 0 FFFFFFFF 1 Input1** | Red: FFFF0000<br>White: FFFFFFFF<br>Black: FF000000<br>Purple: FFFF00FF<br>Blue: FF0000FF<br>Green: FF00FFFF<br><br>**Note**: NHD-200-TX hex format uses 4 characters vs 8 with NHD-110 & 140-TX. Use the format below is using an NHD-200-TX:<br><br>Red: FC00<br>White: FFFF<br>Yellow: FF00<br>Gray: BDEF<br>Green: BB00<br><br>**text_size =** Value between 1-4<br><br>**TX =** Hostname or alias of encoder |
| Command applies to:<br>NHD-110-TX, NHD-140-TX, NHD-200-TX | |

| Enable/Disable Text Overlay | |
|---|---|
| Command structure:<br>**config set device osd <on\|off> <TX>** | |
| Response structure:<br>**config set device osd <on\|off> <TX>** | |
| Command Example: Get list of all Scenes<br>**config set device osd on Input1** | |
| Response Example: Return data<br>**receive: config set device osd on Input1** | |
| Command applies to:<br>NHD-110-TX, NHD-140-TX, NHD-200-TX | |

# 16.  Contacting Technical Support

Should further clarification of the content of this manual, or assistance on troubleshooting be required, please contact WyreStorm technical support.

Phone: UK: +44 (0) 1793 230 343 | ROW: 844.280.WYRE (9973)
Contact Request: http://wyrestorm.com/contact-tech-support

# 17. Document Revision History

| Section | Update |
|---|---|
| **V6.2 – 210921 | September 2021** | |
| Section 15 – Video Stream Text Overlay | Added section to reflect new commands |
| **V6.1.0 – 210107 | January 2021** | |
| All Sections | Updated to include NetworkHD 110 series<br>Updated to include security protocol information |
| **V6.0.0 – 191113 | November 2019** | |
| API Document Formatting | Reformatting of API documentation. |
| **V5.0.0- 190329 | March 2019** | |
| Supported Product Firmware/Software | Added NHD-250-RX versioning.<br>Updated other versions to latest in firmware release. |
| Configuring and Controlling Multi View Layouts | Added NHD-250-RX to opening statement. |
| **V4.0.1- 190227 | February 2019** | |
| Supported Product Firmware/Software | Reverted NHD-600 firmware version back to released version. |
| **V4.0- 190121 | January 2019** | |
| Supported Product Firmware/Software | Updated with firmware versions released in NetworkHD v4.0 |
| **v3.7- 181221 | December 2018** | |
| Supported Product Firmware/Software | Added NHD-140-TX to the list as all firmware versions being supported. |
| **v3.6- 180913 | September 2018** | |
| Controlling Connected Devices via IR | Correct improper capitalization for the word infrared in the command. |
| **v3.5- 180824 | August 2018** | |
| Supported Product Firmware/Software | Updated to reflect new firmware version. |
| Command Overview | Added content to define that devices follow Telnet (RFC 854) protocol. |
| NHD-400 Switching and Audio Control | Section added with the following commands:<br>   Adjusting Audio Volume<br>   Linking TX USB to an RX<br>   Query TX USB to an RX Linking<br>   Link RX Infrared (IR) to TX<br>   Query RX to TX Infrared (IR) Link<br>   Link RX Serial (RS-232) to TX<br>   Query RX to TX Serial (RS-232) Link |
| Controlling Connected Devices via IR | Added content identifying that commands are in Pronto HEX format and added content that these commands support 400s with v0.9.4 or higher to the existing 600 series note. |
| **v3.4- 180629 | June 2018** | |
| Controlling Connected Devices via IR | Added note that this only applies to the NHD 600.<br>Corrected section numbering issue. |
| Rebooting and Restoring Factory Defaults | Added commands to reboot and restore defaults on the CTL. |
| **v3.3- 180518 | April 2018** | |
| Controlling 3rd Party Devices | Added section on how to control 3rd party devices. |
| Various | Changed Source and Encoder to TX | Changed Endpoint and Decoder to RX. |
| **v3.2- 180509 | April 2018** | |
| Supported Product Firmware/Software | Updated FW versions for API status column. |
| Controlling Connected Devices via IR | Added command to control connected devices via IR. |
| **v3.1- 180506 | April 2018** | |

| | |
|---|---|
| All | Restructure to improve readability. |
| Various | Commands added to support the NHD-600 |
| Supported Product Firmware | Added showing all products and firmware versions supported. |

**v2.1 - 171107 | November 2017**

| TX Preview Streams | Added note to step 6 that adjustments are not available to the NHD-400-TX. |
| --- | --- |

**v2.0 - 170320 | March 2017**

| All | Original Release |
| --- | --- |

# Appendix A: JSON Responses

Where a query command generates a JSON response, the return data will be formatted in a readable JSON format. The format includes some special characters that will influence the display formatting of the data within a Terminal emulator application or parsing considerations when parsing return data by a 3rd-party control system. For ease of reading, the examples below will highlight special characters to differentiate them from printable ASCII characters. In a terminal emulator utility, these characters will not usually be visible or printed (unless enabled in settings) but may influence the layout of the text.

JSON Data Special Characters:

| Command Element | Description |
|---|---|
| <CR> | Carriage Return character (Escape notation = \r) (Hex notation = 0D) |
| <LF> | Line Feed character (Escape notation = \n) (Hex notation = 0A) |
| <HT> | Horizontal Tab character (Escape notation = \t) (Hex notation = 09) |

In addition, the example return data below may use references to other command breakdown sections or provide additional information. While not included as part of the actual return data, this information will also be highlighted for ease of reading.

| Query general information for all TX or RX saved in the NHD-000-CTL |
|---|

Command Example:

```
config get devicejsonstring
```

Return data applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

```
device json string:[<CR><LF>
<HT>    {<CR><LF>
<HT>    <HT>    "aliasName" : "SOURCE1",<CR><LF>
<HT>    <HT>    "deviceType" : "Transmitter",<CR><LF>
<HT>    <HT>    "group" : [<CR><LF>
<HT>    <HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    <HT>    "name" : "ungrouped",<CR><LF>
<HT>    <HT>    <HT>    <HT>    "sequence" : 1<CR><LF>
<HT>    <HT>    <HT>    }<CR><LF>
<HT>    <HT>    ],<CR><LF>
<HT>    <HT>    "ip" : "169.254.232.229",<CR><LF>
<HT>    <HT>    "online" : true,<CR><LF>
<HT>    <HT>    "sequence" : 1,<CR><LF>
<HT>    <HT>    "trueName" : "NHD-140-TX-E4CE02102EE1"<CR><LF>
<HT>    },<CR><LF>
<HT>    {<CR><LF>
<HT>    <HT>    "aliasName" : "SOURCE2",<CR><LF>
<HT>    <HT>    "deviceType" : "Transmitter",<CR><LF>
<HT>    <HT>    "group" : [<CR><LF>
<HT>    <HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    <HT>    "name" : "ungrouped",<CR><LF>
<HT>    <HT>    <HT>    <HT>    "sequence" : 1<CR><LF>
<HT>    <HT>    <HT>    }<CR><LF>
<HT>    <HT>    ],<CR><LF>
<HT>    <HT>    "ip" : "169.254.148.121",<CR><LF>
<HT>    <HT>    "online" : true,<CR><LF>
<HT>    <HT>    "sequence" : 2,<CR><LF>
<HT>    <HT>    "trueName" : "NHD-140-TX-E4CE02102EE2"<CR><LF>
<HT>    },<CR><LF>
<HT>    {<CR><LF>
                next device…
<HT>    },<CR><LF>
continued list of devices in order of sequence…
<HT>    {<CR><LF>
<HT>    <HT>    "aliasName" : "SOURCE7",<CR><LF>
<HT>    <HT>    "deviceType" : "Transmitter",<CR><LF>
<HT>    <HT>    "group" : [<CR><LF>
<HT>    <HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    <HT>    "name" : "ungrouped",<CR><LF>
<HT>    <HT>    <HT>    <HT>    "sequence" : 1<CR><LF>
<HT>    <HT>    <HT>    }<CR><LF>
<HT>    <HT>    ],<CR><LF>
<HT>    <HT>    "ip" : "169.254.1.1",<CR><LF>
<HT>    <HT>    "nameoverlay" : true,<CR><LF>
<HT>    <HT>    "online" : true,<CR><LF>
<HT>    <HT>    "sequence" : 7,<CR><LF>
<HT>    <HT>    "trueName" : "NHD-600-TX-D88039E5E401"<CR><LF>
<HT>    }<CR><LF>
]<CR><LF>
```

North America: 844.280.WYRE (9973) | UK/EMEA: +44 (0) 1793 230 343
Email: Support@WyreStorm.com

**Attribute-value pairs:**

| | |
|---|---|
| **aliasName** | Device Alias |
| **deviceType** | Type of device, either Transmitter (TX) or Receiver (RX) |
| **group** | For a Receiver (RX) - this is the display grouping information assigned in NetworkHD Console |
| **name** | Name of display group |
| **sequence** | Order index of the device in the display group |
| **ip** | Device IPv4 address |
| **nameoverlay** | 600 Series TX only – always "true" |
| **online** | Current online status |
| **sequence** | Order index of the device assigned in NetworkHD Console |
| **trueName** | Device Hostname |
| **txName** | RX Only - Currently assigned TX |

Notes:

Order by index using the "sequence" attribute may not be representative of the order in which device data is returned.
RX devices are returned before TX devices.

## Query TX or RX device working parameters

Command Example:

```
config get device info
```

Return data applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

```
devices json info:<CR><LF>
{<CR><LF>
<HT>    "devices" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
                device attribute-pair list – see device specific examples
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
                next device attribute-pair list – see device specific examples
<HT>    <HT>    },<CR><LF>
                continued list of devices…
<HT>    <HT>    {<CR><LF>
                last device attribute-pair list – see device specific examples
<HT>    <HT>    }<CR><LF>

<HT>    ]<CR><LF>
}<CR><LF>
```

Command Example:

```
config get device info DISPLAY1 SOURCE1
```

Return data below applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

```
devices json info:<CR><LF>
{<CR><LF>
<HT>    "devices" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "audio" : [<CR><LF>
<HT>    <HT>    <HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "mute" : false,<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "name" : "lineout1"<CR><LF>
<HT>    <HT>    <HT>    <HT>    }<CR><LF>
<HT>    <HT>    <HT>    ],<CR><LF>
<HT>    <HT>    <HT>    "edid" : "null",<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : true,<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.7.192",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "autoip",<CR><LF>
<HT>    <HT>    <HT>    "mac" : "e4:ce:02:10:7d:f5",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-220-RX-E4CE02107DF5",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "sourcein" : "NHD-140-TX-E4CE02102EE1;",<CR><LF>
<HT>    <HT>    <HT>    "version" : "v2.12.2"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "cbr_avg_bitrate" : 10000,<CR><LF>
<HT>    <HT>    <HT>    "edid" : "00FFFFFFFFFFFF001C450100010000000812010387341780ACF74A3574CB02309484C
210800D1C0B300950090408180810081C03B80023A801871382D40582C450010090000001E011D8018711C1620582C2500C48E21
00009E000000FC005377697463680A202020202020000000FD00163E0E5010000A20202020202001A702031F7246020490111311F
29090401090401090401830100 0065030C0010008C0AD08A20E02D10103E9600138E21000018662156AA51001E30468F33001009
0000001E662150B051001B30407036001009000000 01E283C80A070B0234030203600100900000001A011D80D0721C1620102C2580
C48E2100009E0000000000007C",<CR><LF>
<HT>    <HT>    <HT>    "enc_fps" : 60,<CR><LF>
<HT>    <HT>    <HT>    "enc_gop" : 60,<CR><LF>
<HT>    <HT>    <HT>    "enc_rc_mode" : "vbr",<CR><LF>
<HT>    <HT>    <HT>    "fixqp_iqp" : 25,<CR><LF>
<HT>    <HT>    <HT>    "fixqp_pqp" : 25,<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "169.254.0.254",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : true,<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.85.242",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "fixed",<CR><LF>
<HT>    <HT>    <HT>    "mac" : "e4:ce:02:10:2e:e3",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-140-TX-E4CE02102EE3",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "profile" : "hp",<CR><LF>
<HT>    <HT>    <HT>    "sourcein" : "unknown",<CR><LF>
<HT>    <HT>    <HT>    "transport_type" : "raw",<CR><LF>
<HT>    <HT>    <HT>    "vbr_max_bitrate" : 20000,<CR><LF>
<HT>    <HT>    <HT>    "vbr_max_qp" : 51,<CR><LF>
<HT>    <HT>    <HT>    "vbr_min_qp" : 0,<CR><LF>
<HT>    <HT>    <HT>    "version" : "v1.0.6"<CR><LF>
<HT>    <HT>    }<CR><LF>
<HT>    ]<CR><LF>
}<CR><LF>
```

**Attribute-value pairs:**

|  |  |
|---|---|
|  | Device Alias |
| **aliasname** | EDID table as hexadecimal string. Set at factory. Could be assigned "null" for default |
| **edid** | Current IPv4 address for default gateway parameter, set using Console utility |
| **gateway** | Shows HDCP compatibility on/off. Set at factory |
| **hdcp** | Current IPv4 address for device network interface. Set using Console utility |
| **ip4addr** | Current device network interface mode – static \| autoip \| dhcp, set using Console |
| **ip_mode** | utility |
| **mac** | Device MAC address. Set at factory |
| **name** | Device Hostname |
| **netmask** | Current subnet mask for device network interface. Set using Console utility |
| **sourcein** | Shows either an assigned TX Hostname (RX) or input port type (TX). Could be |
| **version** | "unknown" |
|  | Shows current device firmware version |

**RX only:**
**audio**

|  |  |
|---|---|
| **mute** | RX audio states |
| **name** | Shows status of mute function |
|  | Shows output port for line audio |

**TX Only:**

|  |  |
|---|---|
| **cbr_avg_bitrate** |  |
| **enc_fps** | CBR mode is not used |
| **enc_gop** | Current video timing stream output frames-per-second |
| **enc_rc_mode** | Shows current range for I-frame in Group of Pictures |
| **fixqp_iqp** | Encoding mode. VBR set at factory |
| **fixqp_pqp** | FixQP mode is not used |
| **profile** | FixQP mode is not used |
| **transport_type** | Encoding profile. High Profile. Set at factory |
| **vbr_max_bitrate** | Transport encapsulation type. "raw" (proprietary). Set at factory |
| **vbr_max_qp** | Maximum bitrate target for VBR encoding. Set using Console utility |
| **vbr_min_qp** | Value for VBR encoding maxQP. Set at factory |
|  | Value for VBR encoding minQP. Set at factory |

```
devices json info:<CR><LF>
{<CR><LF>
        "devices" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "edid" : "null",<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "",<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.6.107",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "dhcp",<CR><LF>
<HT>    <HT>    <HT>    "km_over_ip_enable"  :  "true",<CR><LF>
<HT>    <HT>    <HT>    "mac"  :  "e4:ce:02:10:2e:f0",<CR><LF>
<HT>    <HT>    <HT>    "name" : NHD-400-RX-E4CE02102EF0",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "version" : "v0.10.1",<CR><LF>
<HT>    <HT>    <HT>    "videodetection" : "lost"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "audio_input_type" : "auto",<CR><LF>
<HT>    <HT>    <HT>    "edid" : "null",<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "169.254.0.254",<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.5.209",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "autoip",<CR><LF>
<HT>    <HT>    <HT>    "km_over_ip_enable"  :  "true",<CR><LF>
<HT>    <HT>    <HT>    "mac"  :  "e4:ce:02:10:6e:9e",<CR><LF>
<HT>    <HT>    <HT>    "name" : NHD-400-TX-E4CE02106E9E",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "version" : "v0.10.1",<CR><LF>
<HT>    <HT>    <HT>    "videodetection" : "lost"<CR><LF>
<HT>    <HT>    }<CR><LF>
<HT>    ]<CR><LF>
}<CR><LF>
```

**Attribute-value pairs:**

| | |
|---|---|
| | Device Alias |
| aliasname | EDID table as hexadecimal string. Set at factory. Could be assigned "null" for |
| edid | default |
| gateway | Current IPv4 address for default gateway parameter, set using Console utility |
| ip4addr | Current IPv4 address for device network interface, set using Console utility |
| ip_mode | Current device network interface mode – static \| autoip \| dhcp, set using |
| km_over_ip_enable | Console utility |
| mac | KMoIP mode enabled Boolean value |
| name | Device MAC address. Set at factory |
| netmask | Device Hostname |
| version | Current subnet mask for device network interface. Set using Console utility |
| videodetection | Shows current device firmware version |
| | Video detection is not used |

**TX Only:**
audio_input_type

Input type "auto." Set at factory

```
devices json info:<CR><LF>
{<CR><LF>
<HT>    "devices" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "analog_audio_source" : "analog",<CR><LF>
<HT>    <HT>    <HT>    "edid" : "",<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "0.0.0.0",<CR><LF>
<HT>    <HT>    <HT>    "hdmi_audio_source" : "hdmi",<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.38.229",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "dhcp",<CR><LF>
<HT>    <HT>    <HT>    "mac" : "d8:80:39:e5:e5:25",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-600-RX-D88039E5E525",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "serial_param" : "57600-8n1",<CR><LF>
<HT>    <HT>    <HT>    "sinkpower" : {<CR><LF>
<HT>    <HT>    <HT>    <HT>    "cec" : {<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "onetouchplay" : "4004",<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "standby" : "ff36"<CR><LF>
<HT>    <HT>    <HT>    <HT>    },<CR><LF>
<HT>    <HT>    <HT>    <HT>    "mode" : "CEC",<CR><LF>
<HT>    <HT>    <HT>    <HT>    "rs232" : {<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "mode" : "ascii",<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "onetouchplay" : "!POWERON~",<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "param" : "115200-8n1",<CR><LF>
<HT>    <HT>    <HT>    <HT>    <HT>    "standby" : "!POWROFF~"<CR><LF>
<HT>    <HT>    <HT>    <HT>    }<CR><LF>
<HT>    <HT>    <HT>    },<CR><LF>
<HT>    <HT>    <HT>    "temperature" : 38,<CR><LF>
<HT>    <HT>    <HT>    "version" : "3.6.0.0",<CR><LF>
<HT>    <HT>    <HT>    "video_mode" : "fast_switch",<CR><LF>
<HT>    <HT>    <HT>    "video_stretch_type" : "none",<CR><LF>
<HT>    <HT>    <HT>    "video_timing" : "1080P@50"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "analog_audio_direction" : "INPUT",<CR><LF>
<HT>    <HT>    <HT>    "bandwidth_adjust_mode" : 0,<CR><LF>
<HT>    <HT>    <HT>    "bit_perpixel" : 8,<CR><LF>
<HT>    <HT>    <HT>    "color_space" : "RGB",<CR><LF>
<HT>    <HT>    <HT>    "edid" : "00ffffffffffff004dd903f901010101011b0103806c3d780a0dc9a05747982712484c
2108008180a9c0714fb3000101010101010101010108e80030f2705a80b0588a003d624200001e023a801871382d40582c45003d6242000
01e000000fc00534f4e5920545620202a30300a000000fd00173e0e883c000a20202020202020011d020359f05b61605d5e5f621f1014
05130420223c3e1216030711502060165662c0d7f071507503d07bc570400830f00006e030c003000b83c2f00800102030467d85dc
401788001e200cbe305ff01e50f03000006e3060d01011d007251d01e206e2855003d624200001e0000000000000000000000000000
000000000000a5",<CR><LF>
<HT>    <HT>    <HT>    "gateway" : "0.0.0.0",<CR><LF>
<HT>    <HT>    <HT>    "hdcp14_enable" : true,<CR><LF>
<HT>    <HT>    <HT>    "hdcp22_enable" : true,<CR><LF>
<HT>    <HT>    <HT>    "ip4addr" : "169.254.2.228",<CR><LF>
<HT>    <HT>    <HT>    "ip_mode" : "dhcp",<CR><LF>
<HT>    <HT>    <HT>    "mac" : "d8:80:39:e5:e4:01",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-600-TX-D88039E5E401",<CR><LF>
<HT>    <HT>    <HT>    "netmask" : "255.255.0.0",<CR><LF>
<HT>    <HT>    <HT>    "serial_param" : "57600-8n1",<CR><LF>
<HT>    <HT>    <HT>    "stream0_enable" : true,<CR><LF>
<HT>    <HT>    <HT>    "stream0fps_by2_enable" : false,<CR><LF>
<HT>    <HT>    <HT>    "stream1_enable" : true,<CR><LF>
<HT>    <HT>    <HT>    "stream1_scale" : "960x544",<CR><LF>
<HT>    <HT>    <HT>    "stream1fps_by2_enable" : false,<CR><LF>
<HT>    <HT>    <HT>    "temperature" : 42,<CR><LF>
<HT>    <HT>    <HT>    "version" : "3.6.0.0",<CR><LF>
<HT>    <HT>    <HT>    "video_input" : true,<CR><LF>
<HT>    <HT>    <HT>    "video_source" : "hdmi",<CR><LF>
<HT>    <HT>    <HT>    "video_timing" : "1920x1080P@60"<CR><LF>
```

North America: 844.280.WYRE (9973) | UK/EMEA: +44 (0) 1793 230 343
Email: Support@WyreStorm.com

**Attribute-value pairs:**

| | |
|---|---|
| aliasname | Device Alias |
| edid | EDID table as hexadecimal string. Set at factory. Could be "null" for default |
| gateway | Current IPv4 address for default gateway parameter. Set using Console utility |
| ip4addr | Current IPv4 address for device network interface. Set using Console utility |
| ip_mode | Current device network interface mode – static \| dhcp, set using Console utility |
| mac | Device MAC address. Set at factory |
| name | Device Hostname |
| netmask | Current subnet mask for device network interface. Set using Console utility |
| serial_param | Default UART settings. Set using Console utility |
| temperature | Device temperature at core |
| version | Shows current device firmware version |

**RX Only:**

| | |
|---|---|
| analog_audio_source | Shows current analog audio port assignment – analog \| dmix |
| genlock_scaling_resolution | When video_mode=genlock_scaling, shows output resolution |
| hdmi_audio_source | Shows current HDMI audio assignment – hdmi \| dmix \| analog |
| sinkpower | Shows display power proxy commands. Set using Console utility |
|     cec | Shows CEC proxy command data for: |
|         onetouchplay | onetouchplay proxy command data |
|         standby | standby proxy command data |
|     mode | Shows which proxy command types are generated – ALL \| CEC \| RS232 |
|     rs232 | Shows RS-232 proxy command parameters/data for: |
|         mode | Command string interpretation mode – ascii \| hex |
|         onetouchplay | onetouchplay proxy command data |
|         param | Proxy command UART parameters |
|         standby | standby proxy command data |
| video_mode | Current output video mode – genlock \| genlock_scaling \| fast_switch |
| video_stretch_type | When video_mode=fast_switch, shows stretch type – none \| fit_in \| stretch_out |
| video_timing | When video_mode=fast_switch, shows output timing |

**TX Only:**

| | |
|---|---|
| analog_audio_direction | Shows direction of analog audio port – input \| output |
| bandwidth_adjust_mode | Shows the mode for dynamic bandwidth adjustment (Multiview) – 0 \| 1 \| 2 |
| bit_perpixel | Shows the current input video bit per component / pixel |
| color_space | Shows the current input video colorspace |
| hdcp14_enable | Shows if HDCPv1.x is enabled – true \| false. Set using Console utility |
| hdcp22_enable | Shows if HDCPv2.2 is enabled – true \| false. Set using Console utility |
| stream0_enable | Shows if the main video stream is enabled – true \| false |
| stream0fps_by2_enable | Shows if the input video framerate is halved for the main stream – true \| false |
| stream1_enable | Shows if the Multiview stream is enabled – true \| false |
| stream1_scale | Shows the video output resolution for the Multiview stream |
| stream1fps_by2_enable | Shows if the input video framerate is halved for the Mutiview stream – true \| false |
| video_input | Shows if a valid video input signal is present – true \| false |
| video_source | Shows the current input port assignment – auto \| hdmi \| dp, set using Console |
| video_timing | Shows the current video input timing |

## Query TX or RX device real-time status

Command Example:

```
config get device status
```

Return data applies to:

NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

NHD-400-TX, NHD-400-RX

NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX, NHD-610-TX

```
devices status info:<CR><LF>
{<CR><LF>
<HT>      "devices status" : [<CR><LF>
<HT>      <HT>      {
                    device attribute-pair list – see device specific examples
<HT>      <HT>      },<CR><LF>
<HT>      <HT>      {<CR><LF>
                    next device attribute-pair list – see device specific examples
<HT>      <HT>      },<CR><LF>
                    continued list of devices…
<HT>      <HT>      {<CR><LF>
                    last device attribute-pair list – see device specific examples
<HT>      <HT>      }<CR><LF>

<HT>      ]<CR><LF>
}<CR><LF>
```

Command Example:

```
config get device status DISPLAY1 SOURCE1
```

Return data below applies to:
NHD-110-TX/RX, NHD-100-TX, NHD-140-TX, NHD-100-RX, NHD-200-TX, NHD-200-RX, NHD-210-RX, NHD-220-RX, NHD-250-RX

```
devices status info:<CR><LF>
{<CR><LF>
        "devices status" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "audio stream ip address" : "224.48.46.225",<CR><LF>
<HT>    <HT>    <HT>    "encoding enable" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "line out audio enable" : "false",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-140-TX-E4CE02102EE1",<CR><LF>
<HT>    <HT>    <HT>    "resolution" : "1920x1080",<CR><LF>
<HT>    <HT>    <HT>    "stream frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "stream resolution" : "1920x1080",<CR><LF>
<HT>    <HT>    <HT>    "video stream ip address" : "224.16.46.225"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "audio bitrate" : "1536000",<CR><LF>
<HT>    <HT>    <HT>    "audio input format" : "lpcm",<CR><LF>
<HT>    <HT>    <HT>    "hdcp status" : "hdcp14",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out audio enable" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out resolution" : "1920x1080",<CR><LF>
<HT>    <HT>    <HT>    "line out audio enable" : "true",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-210-RX-E4CE02107132",<CR><LF>
<HT>    <HT>    <HT>    "stream error count" : "0",<CR><LF>
<HT>    <HT>    <HT>    "stream frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "stream resolution" : "1920x1080"<CR><LF>
<HT>    <HT>    }<CR><LF>
<HT>    ]<CR><LF>
}<CR><LF>
```

**Attribute-value pairs:**

| | |
|---|---|
| aliasname | Device Alias |
| line out audio enable | Shows if analog audio output is enabled – true \| false |
| name | Device Hostname |
| stream frame rate | Shows current framerate of incoming video stream |
| stream resolution | Shows current resolution of incoming video stream |

**RX only:**

| | |
|---|---|
| audio  bitrate | Current stream bitrate of audio channel |
| audio input format | Current audio format of incoming audio stream |
| hdcp status | Current HDCP status/version of incoming stream |
| hdmi out active | Current status of HDMI output – true \| false |
| hdmi out audio enable | Shows if HDMI audio output is enabled – true \| false |
| hdmi out frame rate | Current HDMI output video frame rate |
| hdmi out resolution | Current HDMI output video resolution |
| stream error count | Shows number of stream errors since power on |

**TX Only:**

| | |
|---|---|
| audio stream ip address | Current audio stream multicast group IP address |
| encoding enable | Shows status of stream encoder – true \| false |
| hdmi in active | Current status of HDMI input, for valid HDMI signal – true \| false |
| hdmi in frame rate | Current HDMI input video frame rate |
| resolution | Current HDMI input video resolution |
| video stream ip address | Current video stream multicast group IP address |

```
devices status info:<CR><LF>
{<CR><LF>
        "devices status" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "audio bitrate" : "3072000",<CR><LF>
<HT>    <HT>    <HT>    "audio output format" : "lpcm",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : "hdcp22",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out resolution" : "1920x1080",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-400-RX-E4CE02103CB3"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "audio bitrate" : "3072000",<CR><LF>
<HT>    <HT>    <HT>    "audio input format" : "lpcm",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : "hdcp22",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-400-TX-E4CE0210B6D4",<CR><LF>
<HT>    <HT>    <HT>    "resolution" : "1920x1080"<CR><LF>
<HT>    <HT>    }<CR><LF>
<HT>    ]<CR><LF>
}<CR><LF>
```

**Attribute-value pairs:**

| | |
|---|---|
| aliasname | Device Alias |
| audio bitrate | Current stream bitrate of audio channel |
| hdcp | Current HDCP status/version of AV stream |
| name | Device Hostname |

**RX only:**

| | |
|---|---|
| audio output format | Current HDMI audio output format |
| hdmi out active | Current status of HDMI output – true \| false |
| hdmi out frame rate | Current HDMI output video frame rate |
| hdmi out resolution | Current HDMI output video resolution |

**TX only:**

| | |
|---|---|
| audio input format | Current audio format of HDMI input |
| hdmi in active | Current status of HDMI input, for valid HDMI signal – true \| false |
| hdmi in frame rate | Current HDMI input video frame rate |
| resolution | Current HDMI input video resolution |

Return data below applies to:
NHD-600-TX, NHD-600-RX, NHD-600-TRX, NHD-600-TRXF, NHD-610-RX

```
devices status info:<CR><LF>
{<CR><LF>
        "devices status" : [<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "DISPLAY1",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : "hdcp14",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "hdmi out resolution" : "1920x1080",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-600-RX-D88039E5E525"<CR><LF>
<HT>    <HT>    },<CR><LF>
<HT>    <HT>    {<CR><LF>
<HT>    <HT>    <HT>    "aliasname" : "SOURCE1",<CR><LF>
<HT>    <HT>    <HT>    "hdcp" : "hdcp14",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in active" : "true",<CR><LF>
<HT>    <HT>    <HT>    "hdmi in frame rate" : "60",<CR><LF>
<HT>    <HT>    <HT>    "name" : "NHD-600-TX-D88039E5ED1E",<CR><LF>
<HT>    <HT>    <HT>    "resolution" : "1920x1080"<CR><LF>
<HT>    <HT>    }<CR><LF>
<HT>    ]<CR><LF>
}<CR><LF>
```

Attribute-value pairs:

| | |
|---|---|
| **aliasname** | Device Alias |
| **hdcp** | Current HDCP status/version of AV stream |
| **name** | Device Hostname |

**RX only:**

| | |
|---|---|
| **hdmi out active** | Current status of HDMI output – true \| false |
| **hdmi out frame rate** | Current HDMI output video frame rate |
| **hdmi out resolution** | Current HDMI output video resolution |

**TX only:**

| | |
|---|---|
| **hdmi in active** | Current status of HDMI input, for valid HDMI signal – true \| false |
| **hdmi in frame rate** | Current HDMI input video frame rate |
| **resolution** | Current HDMI input video resolution |

# Publication Disclaimer

The material contained in this document consists of information that is the sole property of WyreStorm. This document is intended to provide information to allow interfacing to the relevant WyreStorm equipment by third party products.

WYRESTORM IS NOT RESPONSIBLE FOR MALFUNCTIONS AND/OR THE IN-OPERABILITY WHICH MAY BE CAUSED BY THE APPLICATION OF THIS INFORMATION, WHETHER EXPECTED OR NOT.

WyreStorm reserves the right to change software, control codes, and specifications without notice.

WyreStorm will not be liable for any use of this information or any changes it may make to those products. The use of this information constitutes an agreement by the user to these limitations and exclusions.