



DigitalinxIP Programming Guide / API

API Version 1.17

Table of Contents

API Overview	4
LAN Connections	4
RS232 Connections.....	4
Terminology	5
API Commands.....	5
config Commands	6
[config set] Commands Summary	6
config set ip4addr	7
config set ip4addr2	8
config set webloginpasswd	9
config set restorefactory	10
config set reboot.....	10
config set device alias	11
config set device remove.....	12
config set device ip.....	13
config set device reboot.....	14
config set device restorefactory	15
config set device info.....	16
config set device info - parameters and values	17
config set device sinkpower	18
config set device audio input type - 5000 Only	19
config set device audio volume - 5000 Only.....	20
config set device status notify.....	21
config set session alias.....	22
config set telnet alias	22
config set RS232 alias	23
config set service sshservice.....	23
config get Commands Summary	24
config get version.....	24
config get devicelist	25
config get ipsetting	25
config get ipsetting2	26
config get name	27
config get device info	28
config get device info - parameters and values	29
config get devicejsonstring	30
config get scenejsonstring.....	33
config get telnet alias.....	41
config get rs-232 alias.....	41
config get system sshservice	42

matrix Commands	43
<i>matrix set Commands Summary</i>	43
<i>matrix set</i>	43
<i>matrix video set (5000 series only)</i>	44
<i>matrix audio set (5000 series only)</i>	45
<i>matrix usb set (5000 series only)</i>	46
<i>matrix infrared set (5000 series only)</i>	47
<i>matrix serial set (5000 series only)</i>	48
<i>matrix get Commands Summary</i>	49
<i>matrix get</i>	49
<i>matrix video get</i>	50
<i>matrix audio get</i>	51
<i>matrix usb get</i>	52
<i>matrix infrared get</i>	53
<i>matrix serial get</i>	54
vw (Video Wall) Commands	55
<i>vw Commands Summary</i>	55
<i>vw add</i>	55
<i>vw rm (video wall)</i>	56
<i>vw rm (devices)</i>	57
<i>vw add position</i>	58
<i>vw add layout</i>	59
<i>vw change (decoder - encoder)</i>	60
<i>vw change (video wall - encoder)</i>	61
<i>vw bezelgap (5000 series only)</i>	62
<i>vw get</i>	63
serial Command	64
scene Commands	66
<i>scene Commands Summary</i>	66
<i>scene get</i>	66
<i>scene active</i>	67
<i>scene set</i>	67
<i>scene change (scene - encoder)</i>	68
<i>scene connect (scenename)</i>	69
notify Commands	70
<i>notify Commands Summary</i>	70
<i>notify endpoint</i>	70
<i>notify serialinfo</i>	71
<i>notify video (5000 series only)</i>	72
infrared Command	73

API Overview

The API command feedback supports alias names for easy system programming and management. In order to use alias names to identify devices, transmit the command `config set session alias on`. After this command is initiated, all transmissions and responses will use the aliases of the encoders and decoders that you assigned to them. If the IPEXCB is powered off or has to be rebooted, the command will need to be initiated again to properly interact with the alias names of the encoders and decoders in the programming logic.

LAN Connections

The IPEXCB has two Ethernet ports: LAN1 (AV/PoE) and LAN2 (CTRL). It listens to TCP port 23 on both ports, through which you can control and manage the IP video matrix with the API commands in this document.

LAN1 (AV/PoE) is used to connect the IPEXCB to the A/V LAN or network switch where all the DigitalinXIP encoders and decoders reside. The default IP address for LAN1 of the IPEXCB is 169.254.1.1.

LAN2 (CTRL) is used to connect the IPEXCB to a third party control system or network for TCP/IP control. The default IP address for LAN2 of the IPEXCB is 192.168.11.243.

RS232 Connections

The IPEXCB features two RS232 connections labeled *Debug* and *Control*. The *Debug* connection will only communicate with the IPEXCB and will not control any encoders or decoders. The *Control* connections will communicate with all DigitalinXIP encoders and decoders that reside on the A/V LAN or network switch.

To use the RS232 control transport capabilities of the IPEXCB, connect the TX, RX, and ground control signal wires to the middle RS232 connections on the removable 6-pole terminal block. Consult the manual of the 3rd party control system to determine which pins the TX and RX signals are carried on. Be sure to always connect TX to RX and RX to TX.



The RS232 control ports require a standard straight-through serial cable for operation. The default settings for the RS232 ports are:

- Debug connection: 115200 baud, 8 Data Bits, 1 Stop Bit, Parity = none
- Control connection: 9600 baud, 8 Data Bits, 1 Stop Bit, Parity = none

While the IPEXCB requires RS232 commands set at 9600 bps, multiple baud rates are available to communicate with the remote devices connected to the RS232 ports of any DigitalinXIP encoders or decoders.

Terminology

Below is a list of the terminology used in this document with a description of its use.

Terminology	Description
Device	DigitalinxIP Encoder or decoder
Online	Device is working properly and can be controlled by the IPEXCB
Offline	Device cannot be controlled by the IPEXCB for some reason, such as loss of power or disconnected from the Ethernet switch
Hostname	Default name of the device (device type-MAC address), such as IPEX2002-341B22FFFFB3
Alias	Name assigned to device for easy management; only alphanumeric characters and hyphens (-) are supported.

API Commands

Below is a list of the primary commands used in this document with a description of its use.

Commands	Description
config	Manage and configure IPEXCB and devices
matrix	Control the switching of devices
vw	Manage and configure video wall applications
serial	Transmit commands to sources and displays via serial ports on devices
scene	Manage and configure video wall layouts
notify	Inform third party control device of serial response and online status

Below is a list of the secondary commands used in this document with a description of its use.

Commands	Description
set	Applies a change to a primary command
get	Queries the active state via primary command

config Commands

[config set] Commands Summary

Commands	Description
<code>config set ip4addr</code>	Configures network settings for LAN1 (AV/PoE)
<code>config set ip4addr2</code>	Configures network settings for LAN2 (CTRL)
<code>config set webloginpasswd</code>	Changes web GUI login password
<code>config set telnetpasswd</code>	Changes Telnet login password
<code>config set delete telnetpasswd</code>	Deletes current Telnet login password
<code>config set restorefactory</code>	Resets IPEXCB to factory defaults
<code>config set reboot</code>	Reboots IPEXCB
<code>config set device alias</code>	Renames a device
<code>config set device remove</code>	Removes a device record from IPEXCB
<code>config set device ip</code>	Configures device network settings
<code>config set device reboot</code>	Reboots a device
<code>config set device restorefactory</code>	Resets a device to factory defaults
<code>config set device info</code>	Changes device working parameters
<code>config set device sinkpower {on off}</code>	Multiple display devices connected to decoder enter or exit standby
<code>config set device audio input type {hdmi analog}</code>	Configures device audio input type
<code>config set device audio volume {up down} analog</code>	Configures the analog audio volume up or down
<code>config set device status notify {on off}</code>	Sets the status notify system for a device either on (wake) or off (standby)
<code>config set session alias {on off}</code>	Enter or exit the alias mode in current session
<code>config set telnet alias {on off}</code>	Enter or exit the alias mode in current telnet session
<code>config set rs-232 alias {on off}</code>	Enter or exit the alias mode in current RS232 session
<code>config set service sshservice {on off}</code>	Opens or closes the SSH service of the system

config set ip4addr

Configures network settings for LAN1 (AV/PoE) interface on the IPEXCB. This command is used to set the IP address, subnet mask and gateway for the LAN1 (AV/PoE) port. This port only supports static IP mode.

The IPEXCB will automatically reboot after the response for the settings to take effect.

Command Structure

```
config set ip4addr {IP} netmask {N} gateway {G}
```

{IP} = IP Address; xxx.xxx.xxx.xxx

{N} = Netmask; xxx.xxx.xxx.xxx

{G} = Gateway; xxx.xxx.xxx.xxx

Example

Set the LAN1 (AV/PoE) port's IP address to 169.254.1.254, subnet mask to 255.255.0.0, and gateway to 169.254.1.1.

Command

```
config set ip4addr 169.254.1.254 netmask 255.255.0.0 gateway 169.254.1.1
```

Response

```
ip setting will change to: ipaddr 169.254.1.254 netmask 255.255.0.0 gateway 169.254.1.1
```

config set ip4addr2

Configures network settings for LAN2 (CTRL) interface on the IPEXCB. This command is used to set the IP address, subnet mask and gateway for the LAN2 (CTRL) port. This port only supports static IP mode.

The IPEXCB will automatically reboot after the response for the settings to take effect.

Command Structure

```
config set ip4addr2 {IP} netmask {N} gateway {G}
```

```
{IP} = IP Address; xxx.xxx.xxx.xxx
```

```
{N} = Netmask; xxx.xxx.xxx.xxx
```

```
{G} = Gateway; xxx.xxx.xxx.xxx
```

Example

Set the LAN2 (CTRL) port's IP address to 192.168.11.243, subnet mask to 255.255.255.0, and gateway to 192.168.11.1.

Command

```
config set ip4addr2 192.168.11.243 netmask 255.255.0.0 gateway 192.168.11.1
```

Response

```
ip setting2 will change to: ipaddr 192.168.11.243 netmask 255.255.0.0  
gateway 192.168.11.1
```


config set webloginpasswd

Changes the web GUI login password for the IPEXCB

Command Structure

```
config set webloginpasswd {P}
```

{P} = User defined password

Example

Change the web GUI login password to 123456.

Command

```
config set webloginpasswd 123456
```

Response

```
password for web modified
```

config set telnetpasswd

Changes the telnet password on the IPEXCB

Command Structure

```
config set telnetpasswd {P}
```

{P} = User defined password

Example

Change the telnet password to 123456.

Command

```
config set telnetpasswd 123456
```

Response

```
password for telnet modified
```

config set delete telnetpasswd

Deletes the telnet password on the IPEXCB

Command

config set delete telnetpasswd

Response

password for telnet deleted

config set restorefactory

Restores the IPEXCB to factory defaults.

Note: The IPEXCB will automatically reboot after the response for the settings to take effect.

Command

config set restorefactory

Response

system will restore to factory settings now

config set reboot

Reboots the IPEXCB.

Command

config set reboot

Response

system will reboot now

config set device alias

Adds an easy alias description to devices for easier string identification and management using the API commands in this manual. An alias can be used in any other command to replace the original host name. Each alias should be unique. Only alphanumeric characters and hyphens (-) may be used in the device name.

Command Structure

```
config set device alias {I} {A}
```

{I} = default device identifier

{A} = desired alias for device

Example

Rename IPEX2002-341B22FFFFB3 to alias named DVD1.

Command

```
config set device alias IPEX2002-341B22FFFFB3 DVD1
```

Response

```
IPEX2002-341B22FFFFB3's alias is DVD1
```

config set device remove

Removes a device record for an encoder and decoder from the IPEXCB. One or more device records may be removed at the same time. When a device's record is removed, it cannot be detected or controlled by the IPEXCB. To restore the removed online device, reboot the device or the IPEXCB controller.

Command Structure

```
config set device remove {A1} {A2}
```

```
{A1} = alias of device 1  
{A2} = alias of device 2
```

Example

Remove the devices with the alias name of IN1 and OUT2

Command

```
config set device remove IN1 OUT2
```

Response

the following device's record will be removed:

```
IN1  
OUT2
```

config set device ip

Configures device network settings. Devices support Auto IP, DHCP and Static IP for network configuration. When setting devices *Static* mode, the IP address, subnet mask and gateway must be all be declared at the same time. The device will need to be manually rebooted for the settings to take effect.

Command Structure

```
config set device ip {A} {Mode}
```

{A} = Alias of device

{Mode} = [autoip|dhcp|static ip4addr netmask gateway]

Example

Sets decoder DISPLAY1 (alias) to to Static IP (IP address 169.254.5.253, subnet mask 255.255.0.0, gateway 169.254.1.253).

Command

```
config set device ip DISPLAY1 static 169.254.5.253 255.255.0.0 169.254.1.253
```

Response

Devices's ipsetting will change to:

```
IPEX5002-341B22800BCA static 169.254.5.253 255.255.0.0 169.254.1.253
```

Note: multiple addresses can be set for multiple devices all at once by separating each device name and desired IP mode with a comma

```
config set device ip {A1} {Mode} {A2} {Mode}
```

{A1}{A2} = Alias of devices

{Mode} = [autoip|dhcp|static ip4addr netmask gateway]

config set device reboot

Reboots one or more devices.

Command Structure

```
config set device reboot {A}
```

{A} = Alias of device

Example

Reboot decoder device with the alias name of DISPLAY1

Command

```
config set device reboot DISPLAY1
```

Response

```
the following device will reboot now:  
DISPLAY1
```

Note: multiple devices can be rebooted all at once by using the following command structure

```
config set device reboot {A1} {A2} {A3}
```

{A1}{A2}{A3} = Alias of devices

config set device restorefactory

Resets one or more devices to the factory default settings. The device will automatically reboot after the response for the settings to take effect.

Command Structure

```
config set device restorefactory {A}
```

{A} = Alias of device

Example

Resets device with the alias name of IN1 to factory defaults

Command

```
config set device restore IN1
```

Response

```
the following device will restore to factory setting now:  
IN1
```

Note: multiple devices can be set to factory default all at once by separating each device name with a comma

```
config set device restorefactory {A1} {A2} {A3}
```

{A1}{A2}{A3} = Alias of devices

config set device info

Changes the devices working parameters, such as bandwidth utilization or IP settings. One or more devices plus one or more settings may be configured at the same time. The device will need to be manually rebooted for the settings to take effect. A list of supported parameters with values is on the next page that can be used with this command structure.

Command Structure

```
config set device info {P1=value1} {P2=value2} {A}
```

{P1=value1} = Parameter1 name and parameter1 value

{P2=value2} = Parameter2 name and parameter2 value

{A} = Alias of device

Example

Sets device with the alias name of DISPLAY1 to DHCP mode

Command

```
config set device info ip_mode=dhcp DISPLAY1
```

Response

```
config set device info ip_mode=dhcp DISPLAY1
```


config set device info - parameters and values

Send data in ***parameter-value*** format. The following table lists the parameters supported by devices and their value ranges. All parameters can be changed, unless otherwise stated as **READ ONLY**

Parameter	Description / Value Range	Device
name	READ ONLY. Device name. Format is "Device type-MAC address" such as IPEX2001-341B22FFFFB3.	All devices
version	READ ONLY. Device software version. Format is v#.## such as v2.5.6.	All devices
mac	READ ONLY. Device MAC address.	All devices
ip_mode	IP address mode autoip: AutoIP static: Static IP dhcp: DHCP	All devices
ip4addr	IPv4 address. When ip_mode is static, IPv4 address can be changed.	All devices
netmask	IPv4 subnet mask. When ip_mode is static, IPv4 subnet mask can be changed.	All devices
gateway	IPv4 gateway. When ip_mode is static, IPv4 gateway can be changed.	All devices
enc_rc_mode	Data rate control method. cbr is CBR mode. vbr is VBR mode. fixqp is Fixed QP mode.	2000 encoder only
profile	Encoding profile. bp is base profile. mp is main profile. hp is high profile.	Encoder
cbr_avg_bitrate	CBR encoding average rate. Unit is kbps. Data rate of IPEX2001 is less than or equal to 30720.	2000 encoder only
vbr_max_bitrate	VBR encoding maximum rate. Unit is kbps. Data rate of IPEX2001 is less than or equal to 30720.	2000 encoder only
vbr_min_gp	VBR minimum quantification parameters. Value range is [0 - 51].	2000 encoder only
vbr_max_gp	VBR maximum quantification parameters. Value range is [0 - 51].	2000 encoder only
fixqp_iqp	FixQP encoding mode I-frame quantification parameters. Value range is [0 - 51].	2000 encoder only
fixqp_pqp	FixQP encoding mode P-frame quantification parameters. Value range is [0 - 51].	2000 encoder only
enc_gop	GOP size. Value range is [1 - 65535]. There is one I-frame in a specific range.	2000 encoder only
enc_fps	Frames per second. Value range is [1 - 60].	2000 encoder only
transport_type	Streaming media encapsulation format. raw is private format. ts is MPEG-2 TS format.	Encoder
audio.name	READ-ONLY. Audio interface name. Names like linein1, linein2, lineout1 and lineout2 are related to device hardware configuration.	All devices
audio.mute	Audio interface mute status. true is "mute". false is "unmute". For example, audio.lineout1.mute=true.	Decoder

config set device sinkpower

Transmit either the CEC or serial ON/OFF command to a display connected to an IPEX5002 decoder via HDMI (CEC) or by serial (RS232).

The communication preference and the ON/OFF serial strings for the decoder must be configured in DigitalinXIP Configurator under the *BATCH COMMANDS > POWER* menu to be operable.

Command Structure

```
config set device sinkpower {P} {A1} {A2}
```

```
{P} = Power status; [on|off]  
{A1} {A2} = Alias of devices
```

Example

Transmits the ON command for two displays connected to two decoders either by CEC (HDMI) or by serial (RS232) communication, each decoders alias is *OUT1* and *OUT2* for this example

Command

```
config set device sinkpower on OUT1 OUT2
```

Response

```
config set device sinkpower on OUT1 OUT2
```

config set device audio input type - 5000 Only

Defines which audio input stream will be used in an encoder, either embedded HDMI digital audio or the analog audio input

Command Structure

```
config set device audio input type {T} {A}
```

{T} = Input selection; [hdmi|analog|auto]

{A} = Alias of device

Example

Sets the default encoder audio input of device named IN1 to analog audio

Command

```
config set device audio input type analog IN1
```

Response

```
config set device audio input type analog IN1
```

config set device audio volume - 5000 Only

Configures the analog audio volume up or down on an encoder or decoder

Command Structure

```
config set device audio volume {V} analog {A}
```

{V} = Volume adjustment; [up|down]
{A} = Alias of device

Example

Turns the volume adjustment up on decoder with alias name of IN1

Command

```
config set device audio volume up analog IN1
```

Response

```
config set device audio volume up analog IN1
```

config set device status notify

Sets notify status for a device or group of devices to either ON or OFF

Command Structure

```
config set device status notify {N} {A1} {A2}
```

```
{N} = Notify status; [on|off]  
{A1} {A2} = Alias of device
```

Example

Sets notify status ON for an encoder devices with the alias name IN1 and IN2

Command

```
config set device status notify on IN1 IN2
```

Response

```
config set device status notify on IN1 IN2
```

Note: status notify can be set in groups of devices

Command Structure

```
config set device status notify {N} {G}
```

```
{N} = Notify status; [on|off]  
{G} = G
```

Example

Sets notify status ON for all devices

Command

```
config set device status notify on ALL_DEV
```

Response

```
config set device status notify on ALL_DEV
```

config set session alias

Turns on or off alias mode for the current session. If the value is set to *on*, then all API commands will respond with the device alias. If the value is set to *off*, then all API commands will respond with the device name.

Command Structure

```
config set session alias {S}
```

```
{S} = Alias status; [on|off]
```

Example

```
config set session alias on
```

Response

```
config set session alias on
```

config set telnet alias

Turns on or off alias mode for the current telnet session. If the value is set to *on*, then all API commands will respond with the device alias. If the value is set to *off*, then all API commands will respond with the device name.

Command Structure

```
config set telnet alias {S}
```

```
{S} = Alias status; [on|off]
```

Example

```
config set telnet alias on
```

Response

```
config set telnet alias on
```

config set RS232 alias

Turns on or off alias mode for the current telnet session. If the value is set to *on*, then all API commands will respond with the device alias. If the value is set to *off*, then all API commands will respond with the device name.

Command Structure

```
config set rs-232 alias {S}
```

{S} = Alias status; [*on|off*]

Example

```
config set rs-232 alias on
```

Response

```
config set rs-232 alias on
```

config set service sshservice

Opens or closes the SSH service of the system

Command Structure

```
config set service sshservice {S}
```

{S} = SSH service status; [*on|off*]

Example

```
config set service sshservice on
```

Response

```
config set service sshservice on
```

config get Commands Summary

Commands	Description
<code>config get version</code>	Queries IPEXCB version information
<code>config get devicelist</code>	Queries an online device list
<code>config get ipsetting</code>	Queries network settings for LAN1 (AV/PoE) on the IPEXCB
<code>config get ipsetting2</code>	Queries network settings for LAN2 (CTRL) on the IPEXCB
<code>config get name</code>	Queries a device name or its alias
<code>config get device info</code>	Queries device working parameters
<code>config get devicejsonstring</code>	Queries all device information
<code>config get scenejsonstring</code>	Queries all scene information
<code>config get telnet alias</code>	Queries alias mode status for current Telnet session
<code>config get rs-232 alias</code>	Queries alias mode status for current RS232 session
<code>config get system sshservice</code>	Queries SSH service status of system

config get version

Queries the IPEXCB for the current API, web console and firmware revisions.

Command

config get version

Command

config get version

Response

API version: v1.17

System version: LAVS 1.0 v8.2.5 (v8.2.1)

config get devicelist

Queries the network for a list of all active, online devices. The devices that are returned are all decoders followed by all encoders, but there is no separation between the devices. To get a list of all devices, including offline ones, please use *config get devicejsonstring*.

Command

config get devicelist

Command

config get devicelist

Response

devicelist is OUT1 OUT2 IN1 IN2

config get ipsetting

Queries the network settings for LAN1 (AV/PoE) for the IPEXCB

Command

config get ipsetting

Command

config get ipsetting

Response

ipsetting is:ip4addr 169.254.1.1 netmask 255.255.0.0 gateway 169.254.1.254

config get ipsetting2

Queries the network settings for LAN2 (CTRL) for the IPEXCB

Command

config get ipsetting

Command

config get ipsetting2

Response

ipsetting is:ip4addr 192.168.11.243 netmask 255.255.0.0 gateway 192.168.11.1

config get name

Queries a device hostname or its alias. A device name can be used to get an alias and vice versa. If `config get name` is used without parameters, the response is all device names and corresponding aliases.

Note: A device without an alias will respond with `NULL`

Command Structure

```
config get name {H}
```

{H} = hostname of device

Example

Query the alias name assigned to IPEX2001-341B22430115

Command

```
config get name IPEX2001-341B22430115
```

Response

```
IPEX2001-341B22430115's alias is IN1
```

Note: when issuing the `config get name` command without a hostname, a response will be given for ALL online devices

Example

Query all device names and their aliases

Command

```
config get name
```

Response

```
IPEX2002-341B22801761's alias is BottomLeft-2x2  
IPEX2002-341B22801763's alias is TopRight-2x2  
IPEX2002-341B22801768's alias is TopLeft-2x2  
IPEX2002-341B22801769's alias is BottomRight-2x2  
IPEX2001-341B22801758's alias is iPad  
IPEX2001-341B2280175E's alias is PC
```

config get device info

Queries device working parameters, such as bandwidth utilization or IP settings. One or more devices may be queried at the same time. A list of supported device parameters with values is on next page.

Note: if *config get device info* is used without an alias or hostname, info for ALL devices will be returned

Command Structure

```
config get device info {A1} {A2}
```

{A1} {A2} = Alias of Devices

Command

```
config get device info of an IPEX2001 encoder with alias of PC1
```

Response

```
devices json info:
```

```
{
  "devices" : [
    {
      "aliasname" : "PC1",
      "cbr_avg_bitrate" : 10000,
      "enc_fps" : 60,
      "enc_gop" : 60,
      "enc_rc_mode" : "vbr",
      "fixqp_iqp" : 25,
      "fixqp_pqp" : 25,
      "gateway" : "",
      "hdcpc" : true,
      "ip4addr" : "169.254.135.115",
      "ip_mode" : "autoip",
      "mac" : "34:1b:22:80:17:5e",
      "name" : "IPEX2001-341B2280175E",
      "netmask" : "255.255.0.0",
      "profile" : "hp",
      "sourcein" : "hdmi1",
      "transport_type" : "raw",
      "vbr_max_bitrate" : 20000,
      "vbr_max_qp" : 40,
      "vbr_min_qp" : 0,
      "version" : "v2.9.2"
    }
  ]
}
```

config get device info - parameters and values

The following table lists the parameters supported by devices and their value ranges.

Parameter	Description / Value Range	Device
name	Device name. Format is "Device type-MAC address" such as IPEX2001-341B22FFFFB3.	All devices
version	Device software version. Format is v#. #.# such as v2.5.6.	All devices
mac	Device MAC address.	All devices
ip_mode	IP address mode autoip: AutoIP static: Static IP dhcp: DHCP	All devices
ip4addr	IPv4 address. When ip_mode is static, IPv4 address can be changed.	All devices
netmask	IPv4 subnet mask. When ip_mode is static, IPv4 subnet mask can be changed.	All devices
gateway	IPv4 gateway. When ip_mode is static, IPv4 gateway can be changed.	All devices
enc_rc_mode	Data rate control method. cbr is CBR mode. vbr is VBR mode. fixqp is Fixed QP mode.	2000 encoder only
profile	Encoding profile. bp is base profile. mp is main profile. hp is high profile.	Encoder
cbr_avg_bitrate	CBR encoding average rate. Unit is kbps. Data rate of IPEX2001 is less than or equal to 30720.	2000 encoder only
vbr_max_bitrate	VBR encoding maximum rate. Unit is kbps. Data rate of IPEX2001 is less than or equal to 30720.	2000 encoder only
vbr_min_gp	VBR minimum quantification parameters. Value range is [0 - 51].	2000 encoder only
vbr_max_gp	VBR maximum quantification parameters. Value range is [0 - 51].	2000 encoder only
fixqp_iqp	FixQP encoding mode I-frame quantification parameters. Value range is [0 - 51].	2000 encoder only
fixqp_pqp	FixQP encoding mode P-frame quantification parameters. Value range is [0 - 51].	2000 encoder only
enc_gop	GOP size. Value range is [1 - 65535]. There is one I-frame in a specific range.	2000 encoder only
enc_fps	Frames per second. Value range is [1 - 60].	2000 encoder only
transport_type	Streaming media encapsulation format. raw is private format. ts is MPEG-2 TS format.	Encoder
audio.name	Audio interface name. Names like linein1, linein2, lineout1 and lineout2 are related to device hardware configuration.	All devices
audio.mute	Audio interface mute status. true is "mute". false is "unmute". For example, audio.lineout1.mute=true.	Decoder

config get devicejsonstring

Queries all device information for all devices connected to the IPEXCB in a JSON (JavaScript Object Notation) format.

Below is a description of the parameters in the response string:

- ***aliasname*** is the assigned alias for the device; if no alias is shown, the device has not been given an alias.
- ***deviceType*** is the type of device: encoder (transmitter) or decoder (receiver).
- ***group*** is the assigned group of the decoder; a decoder can only be in one group.
 - * ***sequence*** in a *group* is the position of the group, which starts at 1; if the sequence value is 0, the group is not arranged in a specific order.
- ***ip*** is the IP address for the device. ***online*** is the the device activity status: *true* is online, *false* is offline
- ***sequence*** is the position of the decoder within a group, which starts at 1; if the sequence value is 0, the device is not arranged in a specific order.
- ***trueName*** is the devices hostname.
- ***txName (decoder only)*** : Hostname of encoder device that the current decoder is routed to

Command

config get devicejsonstring

Example

Query all device information of which IN1, IN2, OUT1 AND OUT2 are connected to IPEXCB / AV LAN

Command

```
config get devicejsonstring
```

Response

```
device json string:[
  {
    "aliasName" : "OUT1",
    "deviceType" : "Receiver",
    "group" : [
      {
        "name" : "ungrouped",
        "sequence" : 1
      }
    ],
    "ip" : "169.254.4.33",
    "online" : true,
    "sequence" : 4,
    "trueName" : "IPEX2002-341B22801761",
    "txName" : "IPEX2001-341B2280175E"
  },
  {
    "aliasName" : "OUT2",
    "deviceType" : "Receiver",
    "group" : [
      {
        "name" : "ungrouped",
        "sequence" : 2
      }
    ],
    "ip" : "169.254.81.114",
    "online" : true,
    "sequence" : 1,
    "trueName" : "IPEX2001-341B22801763",
    "txName" : "IPEX2001-341B2280175E"
  }
]
```

Response (continued)

```
{
  "aliasName" : "IN1",
  "deviceType" : "Transmitter",
  "group" : [
    {
      "name" : "ungrouped",
      "sequence" : 1
    }
  ],
  "ip" : "169.254.7.33",
  "online" : true,
  "sequence" : 1,
  "trueName" : "IPEX2002-341B22804517",
  "txName" : "IPEX2001-341B2280175E"
},
{
  "aliasName" : "IN2",
  "deviceType" : "Transmitter",
  "group" : [
    {
      "name" : "ungrouped",
      "sequence" : 2
    }
  ],
  "ip" : "169.254.81.114",
  "online" : true,
  "sequence" : 2,
  "trueName" : "IPEX2001-341B22874854",
  "txName" : "IPEX2001-341B2280175E"
}
```


config get scenejsonstring

Queries all scene (layout) information for the IPEXCB.

- ***group*** defines a group. One scene can only be put in one group.
 - * ***sequence*** in ***group*** defines the position of this group , which starts with 1. If the sequence value is 0, it means that this group is not arranged in specific order.
- ***layoutseq*** represents the position of the scene in a video wall.
- ***n*** and ***m*** define the number of rows and columns respectively in a scene.
- ***name*** defines the scene name, such as *s*.
- ***decoderxArray*** defines the decoder position within a two-dimensional array in a scene.
- ***sequence*** in a scene defines the position of a video wall which contains this scene, which starts with 1. If the sequence value is 0, it means that this video wall is not arranged in specific order.
- ***encoderxListArray*** defines the encoder position within a two-dimensional array in a scene.
- ***vwConfigList*** defines the configuration of combination screen in a scene.
 - * ***name*** defines the combination screen name, which uses the naming structure of *scene name_combination screen name* within the IPEXCB.
 - ***pos_row*** defines the start place of the first row.
 - ***pos_col*** defines the start place of the first column.
 - ***row_count*** defines the number of rows in the combination screen.
 - ***col_count*** defines the number of columns in the combination screen.

Command

config get scenejsonstring

Command**`config get scenejsonstring`****Response**

```
scene json string:[
  {
    "group" : [
      {
        "name" : "2x2-Wall",
        "sequence" : 2
      }
    ],
    "layoutseq" : 1,
    "m" : 2,
    "n" : 2,
    "name" : "Matrix-matrix",
    "rxArray" : [
      [
        {
          "aliasName" : "TopLeft-2x2",
          "deviceType" : "Receiver",
          "group" : [
            {
              "name" : "2x2-Wall",
              "sequence" : 2
            }
          ],
          "online" : true,
          "rxstatus" : 1,
          "sequence" : 2,
          "trueName" : "IPEX2002-341B22801768",
          "txName" : "IPEX2001-341B2280175E"
        },
        {
          "aliasName" : "TopRight-2x2",
          "deviceType" : "Receiver",
          "group" : [
            {
              "name" : "2x2-Wall",
              "sequence" : 2
            }
          ],
          "online" : true,
          "rxstatus" : 1,
          "sequence" : 1,
          "trueName" : "IPEX2002-341B22801763",
          "txName" : "IPEX2001-341B2280175E"
        }
      ],
      [
        {
          "aliasName" : "BottomLeft-2x2",
          "deviceType" : "Receiver",
```

```

scene json string:[
  {
    "group" : [
      {
        "name" : "2x2-Wall",
        "sequence" : 2
      }
    ],
    "layoutseq" : 1,
    "m" : 2,
    "n" : 2,
    "name" : "Matrix-matrix",
    "rxArray" : [
      [
        {
          "aliasName" : "TopLeft-2x2",
          "deviceType" : "Receiver",
          "group" : [
            {
              "name" : "2x2-Wall",
              "sequence" : 2
            }
          ],
          "online" : true,
          "rxstatus" : 1,
          "sequence" : 2,
          "trueName" : "IPEX2002-341B22801768",
          "txName" : "IPEX2001-341B2280175E"
        },
        {
          "aliasName" : "TopRight-2x2",
          "deviceType" : "Receiver",
          "group" : [
            {
              "name" : "2x2-Wall",
              "sequence" : 2
            }
          ],
          "online" : true,
          "rxstatus" : 1,
          "sequence" : 1,
          "trueName" : "IPEX2002-341B22801763",
          "txName" : "IPEX2001-341B2280175E"
        }
      ],
      [
        {
          "aliasName" : "BottomLeft-2x2",
          "deviceType" : "Receiver",
          "group" : [
            {
              "name" : "2x2-Wall",
              "sequence" : 2
            }
          ],
          "online" : true,

```

```
        "rxstatus" : 1,
        "sequence" : 4,
        "trueName" : "IPEX2002-341B22801761",
        "txName" : "IPEX2001-341B2280175E"
    },
    {
        "aliasName" : "BottomRight-2x2",
        "deviceType" : "Receiver",
        "group" : [
            {
                "name" : "2x2-Wall",
                "sequence" : 2
            }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 3,
        "trueName" : "IPEX2002-341B22801769",
        "txName" : "IPEX2001-341B2280175E"
    }
]
],
"sceneAutoApply" : false,
"sequence" : 1,
"txListArray" : [
    [
        {
            "devices" : [
                {
                    "aliasName" : "iPad",
                    "deviceType" : "Transmitter",
                    "group" : [
                        {
                            "name" : "ungrouped",
                            "sequence" : 1
                        }
                    ],
                    "online" : true,
                    "rxstatus" : 1,
                    "sequence" : 1,
                    "trueName" : "IPEX2001-341B22801758"
                }
            ]
        },
        {
            "devices" : [
                {
                    "aliasName" : "iPad",
                    "deviceType" : "Transmitter",
                    "group" : [
                        {
                            "name" : "ungrouped",
                            "sequence" : 1
                        }
                    ]
                }
            ]
        }
    ]
]
```

```

    }
    ],
    "online" : true,
    "rxstatus" : 1,
    "sequence" : 1,
    "trueName" : "IPEX2001-341B22801758"
  }
]
},
[
  {
    "devices" : [
      {
        "aliasName" : "PC",
        "deviceType" : "Transmitter",
        "group" : [
          {
            "name" : "ungrouped",
            "sequence" : 1
          }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 2,
        "trueName" : "IPEX2001-341B2280175E"
      }
    ]
  },
  {
    "devices" : [
      {
        "aliasName" : "PC",
        "deviceType" : "Transmitter",
        "group" : [
          {
            "name" : "ungrouped",
            "sequence" : 1
          }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 2,
        "trueName" : "IPEX2001-341B2280175E"
      }
    ]
  }
]
},
{
  "group" : [

```

```
        {
            "name" : "2x2-Wall",
            "sequence" : 2
        }
    ],
    "layoutseq" : 2,
    "m" : 2,
    "n" : 2,
    "name" : "Matrix-2x2VW",
    "rxArray" : [
        [
            {
                "aliasName" : "TopLeft-2x2",
                "deviceType" : "Receiver",
                "group" : [
                    {
                        "name" : "2x2-Wall",
                        "sequence" : 2
                    }
                ],
                "online" : true,
                "rxstatus" : 1,
                "sequence" : 2,
                "trueName" : "IPEX2002-341B22801768",
                "txName" : "IPEX2001-341B2280175E"
            },
            {
                "aliasName" : "TopRight-2x2",
                "deviceType" : "Receiver",
                "group" : [
                    {
                        "name" : "2x2-Wall",
                        "sequence" : 2
                    }
                ],
                "online" : true,
                "rxstatus" : 1,
                "sequence" : 1,
                "trueName" : "IPEX2002-341B22801763",
                "txName" : "IPEX2001-341B2280175E"
            }
        ],
        [
            {
                "aliasName" : "BottomLeft-2x2",
                "deviceType" : "Receiver",
                "group" : [
                    {
                        "name" : "2x2-Wall",
                        "sequence" : 2
                    }
                ],
                "online" : true,
                "rxstatus" : 1,
                "sequence" : 4,
            }
        ]
    ]
}
```

```

        "trueName" : "IPEX2002-341B22801761",
        "txName" : "IPEX2001-341B2280175E"
    },
    {
        "aliasName" : "BottomRight-2x2",
        "deviceType" : "Receiver",
        "group" : [
            {
                "name" : "2x2-Wall",
                "sequence" : 2
            }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 3,
        "trueName" : "IPEX2002-341B22801769",
        "txName" : "IPEX2001-341B2280175E"
    }
]
},
"sceneAutoApply" : false,
"sequence" : 1,
"txListArray" : [
    [
        {
            "devices" : [
                {
                    "aliasName" : "iPad",
                    "deviceType" : "Transmitter",
                    "group" : [
                        {
                            "name" : "ungrouped",
                            "sequence" : 1
                        }
                    ],
                    "online" : true,
                    "rxstatus" : 1,
                    "sequence" : 1,
                    "trueName" : "IPEX2001-341B22801758"
                }
            ]
        },
        {
            "devices" : [
                {
                    "aliasName" : "iPad",
                    "deviceType" : "Transmitter",
                    "group" : [
                        {
                            "name" : "ungrouped",
                            "sequence" : 1
                        }
                    ],
                    "online" : true,

```

```
        "rxstatus" : 1,
        "sequence" : 1,
        "trueName" : "IPEX2001-341B22801758"
    }
  ]
},
[
  {
    "devices" : [
      {
        "aliasName" : "iPad",
        "deviceType" : "Transmitter",
        "group" : [
          {
            "name" : "ungrouped",
            "sequence" : 1
          }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 1,
        "trueName" : "IPEX2001-341B22801758"
      }
    ]
  },
  {
    "devices" : [
      {
        "aliasName" : "iPad",
        "deviceType" : "Transmitter",
        "group" : [
          {
            "name" : "ungrouped",
            "sequence" : 1
          }
        ],
        "online" : true,
        "rxstatus" : 1,
        "sequence" : 1,
        "trueName" : "IPEX2001-341B22801758"
      }
    ]
  }
],
"vwConfigList" : [
  {
    "col_count" : 2,
    "mode" : "0",
    "name" : "2x2vw",
    "oh" : 0,
    "ow" : 0,
    "pos_col" : 0,
    "pos_row" : 0,
    "row_count" : 2,
```


config get telnet alias

Queries telnet alias ON / OFF status

Command

```
config get telnet alias
```

Example

```
config get telnet alias
```

Response

```
telnet alias is on
```

config get rs-232 alias

Queries RS-232 alias ON / OFF status

Command

```
config get rs-232 alias
```

Example

```
config get rs-232 alias
```

Response

```
rs-232 alias is off
```

config get system sshservice

Queries SSH service ON / OFF status

Command

```
config get system sshservice
```

Example

```
config get system sshservice
```

Response

```
system sshservice is on
```

matrix Commands

matrix set Commands Summary

Commands	Description
<code>matrix set</code>	Sets encoder to decoder matrix routes for audio, video, USB and control
<code>matrix video set</code>	Sets independent encoder to decoder video routes (5000 series only)
<code>matrix audio set</code>	Sets independent encoder to decoder audio routes (5000 series only)
<code>matrix usb set</code>	Sets independent encoder to decoder USB routes (5000 series only)
<code>matrix infrared set</code>	Sets independent encoder to decoder IR route (5000 series only)
<code>matrix serial set</code>	Sets independent encoder to decoder RS232 route (5000 series only)

matrix set

Sets the encoder to decoder matrix route for video, audio, usb, IR and serial routes (usb, IR and serial is for 5000 series only). Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using *NULL* as the encoder will stop the video feed going to the decoder. When a decoder is in a video wall, this command is used to switch to another encoder but will not clear video wall settings.

Command Structure

```
matrix set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Route encoder with the alias names of IN1 and IN2 to decoders with the alias name of OUT1 and OUT2

Command

```
matrix set IN1 OUT1, IN2 OUT2
```

Response

```
matrix set IN1 OUT1, IN2 OUT2
```

matrix video set (5000 series only)

Sets an independent encoder to decoder video route for 5000 series products only. Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using `NULL` as the encoder will stop the video feed going to the decoder.

Command Structure

```
matrix video set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Routes the video only from encoders with the alias of IN1 and IN2 to the decoders with the alias name of OUT1 and OUT2

Command

```
matrix video set IN1 OUT1, IN2 OUT2
```

Response

```
matrix video set IN1 OUT1, IN2 OUT2
```

matrix audio set (5000 series only)

Sets the encoder to decoder audio routes for 5000 series products only. Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using `NULL` as the encoder will stop the audio feed going to the decoder.

Command Structure

```
matrix audio set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Routes the audio only from encoders with the alias of IN1 and IN2 to the decoders with the alias name of OUT1 and OUT2

Command

```
matrix audio set IN1 OUT1, IN2 OUT2
```

Response

```
matrix audio set IN1 OUT1, IN2 OUT2
```

matrix usb set (5000 series only)

Sets the encoder to decoder USB routes for 5000 series products only. Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using NULL as the encoder will stop the usb feed going to the decoder. USB routes will only operate in a 1:1 architecture, you cannot connect one USB host of an encoder to multiple decoder clients

Command Structure

```
matrix usb set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Routes USB only from encoders with the alias of IN1 and IN2 to the decoders with the alias name of OUT1 and OUT2

Command

```
matrix usb set IN1 OUT1, IN2 OUT2
```

Response

```
matrix usb set IN1 OUT1, IN2 OUT2
```

matrix infrared set (5000 series only)

Sets the encoder to decoder IR routes for 5000 series products only. Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using `NULL` as the encoder will stop the IR feed going to the decoder. IR can be routed from an encoder to multiple decoders

Command Structure

```
matrix infrared set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Routes IR only from encoders with the alias of IN1 and IN2 to the decoders with the alias name of OUT1 and OUT2

Command

```
matrix infrared set IN1 OUT1, IN2 OUT2
```

Response

```
matrix infrared set IN1 OUT1, IN2 OUT2
```

matrix serial set (5000 series only)

Sets the encoder to decoder serial routes for 5000 series products only. Route groups are separated by commas. Every group starts with the encoder and ends with the decoder(s). Using `NULL` as the encoder will stop the IR feed going to the decoder.

Note: this command facilitates a serial pass through from one device to another. To use this feature properly you must configure the serial settings for each device used in the pass through circuit. For this use the DigitalinXIP Configurator software and right click on the devices and choose *SERIAL* settings. This can also be done in *BATCH COMMANDS* under the *SERIAL* tab.

Command Structure

```
matrix serial set {E1} {D1}, {E2} {D2}
```

{E1}{E2} = Alias of encoder 1 and encoder 2

{D1}{D2} = Alias of decoder 1 and decoder 2

Example

Routes serial only from encoders with the alias of IN1 and IN2 to the decoders with the alias name of OUT1 and OUT2

Command

```
matrix serial set IN1 OUT1, IN2 OUT2
```

Response

```
matrix serial set IN1 OUT1, IN2 OUT2
```


***matrix get* Commands Summary**

Commands	Description
<code>matrix get</code>	Queries encoder to decoder matrix routes
<code>matrix video get</code>	Queries encoder to decoder video routes (5000 series only)
<code>matrix audio get</code>	Queries encoder to decoder audio routes (5000 series only)
<code>matrix usb get</code>	Queries encoder to decoder USB routes (5000 series only)
<code>matrix infrared get</code>	Queries encoder to decoder IR routes (5000 series only)
<code>matrix serial get</code>	Queries encoder to decoder RS232 routes (5000 series only)

matrix get

Queries the encoder to decoder matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A `NULL` as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

`matrix get`

Example

Query current encoder to decoder matrix routes.

Command

`matrix get`

Response

```
matrix information:
IN1 OUT1
IN2 OUT2
null OUT3
null OUT4
```

matrix video get

Queries the encoder to decoder video matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A NULL as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

```
matrix video get
```

Example

Query current encoder to decoder video matrix routes.

Command

```
matrix video get
```

Response

```
matrix information:  
IN1 OUT1  
IN2 OUT2  
null OUT3  
null OUT4
```

matrix audio get

Queries the encoder to decoder audio matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A NULL as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

```
matrix audio get
```

Example

Query current encoder to decoder audio matrix routes.

Command

```
matrix audio get
```

Response

```
matrix information:  
IN1 OUT1  
IN2 OUT2  
null OUT3  
null OUT4
```

matrix usb get

Queries the encoder to decoder USB matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A `NULL` as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

```
matrix usb get
```

Example

Query current encoder to decoder usb matrix routes.

Command

```
matrix usb get
```

Response

```
matrix information:  
IN1 OUT1  
IN2 OUT2  
null OUT3  
null OUT4
```

matrix infrared get

Queries the encoder to decoder IR matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A `NULL` as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

```
matrix infrared get
```

Example

Query current encoder to decoder IR matrix routes.

Command

```
matrix infrared get
```

Response

```
matrix information:  
IN1 OUT1  
IN2 OUT2  
null OUT3  
null OUT4
```

matrix serial get

Queries the encoder to decoder serial matrix routes. Encoder to decoder routes are separated by lines based on the individual decoder. A `NULL` as the encoder indicates no video is going to the decoded. When a decoder is in a video wall, no video wall information is returned.

Command

```
matrix serial get
```

Example

Query current encoder to decoder pass through serial matrix routes.

Command

```
matrix serial get
```

Response

```
matrix information:  
IN1 OUT1  
IN2 OUT2  
null OUT3  
null OUT4
```

vw (Video Wall) Commands

vw Commands Summary

Commands	Description
<code>vw add</code>	Creates an video wall configuration and assigns an encoder
<code>vw rm (video wall)</code>	Removes a video wall configuration
<code>vw rm (devices)</code>	Removes one or more decoders from a video wall configuration
<code>vw add position</code>	Adds one or more decoders to a video wall configuration
<code>vw add layout</code>	Creates an n x m video wall configuration and assigns an encoder and n x m decoders to it
<code>vw change (decoder - encoder)</code>	Removes one decoder from a video wall configuration and sets an encoder to be the new source
<code>vw change (video wall - encoder)</code>	Sets a new encoder as the video wall source
<code>vw bezelgap</code>	Sets bezel compensation parameters (5000 series only)
<code>vw get</code>	Queries all video wall configurations

vw add

`vw add` creates an video wall configuration and assigns an encoder to the video wall. This command is used to create records in the IPEXCB but does not change the devices' working status.

Command Structure

```
vw add {N} {R} {C} {E}
```

```
{N} = Name of video wall
{R} = Number of rows in video wall
{C} = Number of columns in video wall
{E} = Encoder assigned to video wall
```

Example

Create a 2 x 2 video wall configuration `vwtest1` and assign encoder `IPEX2001-341B2243011A`.

Command

```
vw add vwtest1 2 2 IPEX2001-341B2243011A
```

Response

```
videowall item vwtest1 create and assign IPEX2001-341B2243011A to it
```

vw rm (video wall)

vw rm (video wall) removes a video wall configuration. This command is used to remove records in the IPEXCB but does not change the devices' working status. If the current video wall is removed, the active decoders will still play the current content.

Command Structure

vw rm {N}

{N} = Name of video wall

Example

Remove video wall configuration names *vwtest1*.

Command

vw rm vwtest1

Response

videowall item vwtest1 removed

vw rm (devices)

vw rm (devices) removes one or more decoders from a video wall configuration. When the decoder is removed, it shows a full screen image of the current encoder.

Command Structure

vw rm {N} {D}

{N} = Name of video wall
{D} = Decoder assigned to video wall

Example

Remove decoders IPEX5002-341B22800BCE and IPEX5002-341B22800BCA from video wall *vwtest1*.

Command

```
vw rm vwtest1 IPEX5002-341B22800BCE IPEX5002-341B22800BCA
```

Response

```
videowall config change: remove IPEX5002-341B22800BCE IPEX5002-341B22800BCA  
from vwtest1
```

vw add position

vw add position adds one or more decoders to a video wall configuration. Once this command is executed, the decoder will start showing the current video wall content.

Command Structure

```
vw add {N} {D1} {X1} {Y1} {D2} {X2} {Y2}
```

{N} = Name of video wall

{D1}{D2} = Decoder names

{X1}{Y1}{X2}{Y2} = Horizontal / vertical grid positions within video wall for a decoder

Example

Add four decoders with the alias names of OUT1, OUT2, OUT3 and OUT4 to a 2x2 video wall configuration by the name of *vwtest2*.

Command

```
vw add vwtest2 OUT1 1 1 OUT2 1 2 OUT3 2 1 OUT4 2 2
```

Response

```
videowall item vwtest2 configuration added:
```

```
OUT1 1 1
```

```
OUT2 1 2
```

```
OUT3 2 1
```

```
OUT4 2 2
```

vw add layout

vw add layout creates an video wall configuration and assigns an encoder and decoders to it. Once this command is executed, the decoders will start showing the current video wall content.

Decoders are automatically assigned positions in the video wall in the order their names are listed.

DEC11	DEC12	...	DEC1m
DEC21	DEC22	...	DEC2m
...
DECn1	DECn2	...	DECnm

Command Structure

vw add {N} layout {R} {C} {E} {D1} {D2}...

{N} = Name of video wall
 {R} = Number of rows in video wall
 {C} = Number of columns in video wall
 {E} = Encoder assigned to video wall
 {D1}{D2} = Decoders assigned to video wall positions

Example

Create a 2 x 2 video wall configuration *vwtest3* which contains one encoder (alias of IN1) and four decoders that go by the alias of OUT1, OUT2, OUT3 and OUT4

Command

```
vw add vwtest3 layout 2 2 IN1 OUT1 OUT2 OUT3 OUT4
```

Response

```
vw add vwtest3 layout 2 2 IN1 OUT1 OUT2 OUT3 OUT4
```

vw change (decoder - encoder)

vw change (decoder - encoder) clears current video wall layout and removes one decoder from a video wall configuration and sets an encoder to be the new source.

If the encoder is set to `NULL`, no video will be displayed on the decoder.

Command Structure

vw change {D} {E}

{D} = Decoder assigned to video wall
{E} = Newly assigned encoder to removed decoder

Example

Remove decoder with the alias name of `OUT4` from a video wall and switch the decoder to encoder with the alias name of `IN1`

Command

vw change OUT4 IN1

Response

videowall config clear: OUT4 and connect to IN1

vw change (video wall - encoder)

`vw change vw-name encoderx` sets a new encoder as the video wall source.

`vw-name` is the name of the video wall. `encoderx` is the encoder name, which may be the default name or an alias. If the encoder is set to `NULL`, no video will be displayed on the video wall.

Command Structure

vw change {N} {E}

`{N}` = Name of video wall

`{E}` = Newly assigned encoder to video wall

Example

Switch to encoder with the alias of `IN1` for video wall `vwtest2`.

Command

`vw change vwtest2 IN1`

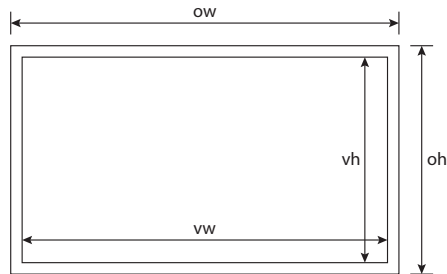
Response

`videowall vwtest2 tx connect to IN1`

vw bezelgap (5000 series only)

`vw bezelgap` sets bezel compensation parameters for 5000 series decoders.

Values for `ow`, `oh`, `vw` and `vh` are in 0.1 mm increments.

***Command Structure***

```
vw bezelgap {N} {ow} {oh} {vw} {vh}
```

```
{N} = Name of video wall  
{ow} = Outside width of display  
{oh} = Outside height of display  
{vw} = Width of display panel  
{vh} = Height of display panel
```

Example

Set bezel compensation parameters with outside size 820 mm (`ow`) x 470 mm (`oh`) and screen size 800 mm (`vw`) x 450 mm (`vh`) in video wall names `vwtest5`.

Command

```
vw bezelgap vwtest5 8200 4700 8000 4500
```

Response

```
videowall vwtest5's bezelgap: 8200 4700 8000 4500
```

vw get

vw get queries all video wall configurations.

Command

vw get

Example

Query all video wall configurations for the connected IPEXCB.

Command

vw get

Response

*Video wall information:
2x2_vw IN1
Row 1: OUT1 OUT2
Row 2: OUT3 OUT4*

serial Command

The **serial** command sends 3rd party device control commands to peripheral devices connected to encoders and decoders via the serial ports. To use this format the serial settings for the devices must be set to 'feed-back' mode. To use this feature properly, use DigitalinXIP Configurator and right click on the devices and choose *SERIAL* settings then choose *FEEDBACK* mode.

Command Structure

```
serial -b {S} -r {R} {H} "{STRING}" {A}
```

{S} = Used to set serial setting of 3rd party device which contains baud rate, data bits, parity and stop bits. Default is **115200-8n1**. (Baud rate is 115200 bps, data bits are 8 bits, parity is "none", stop bits is "1".)

{R} = {on|off} Used to set carriage return at the end of the command, default value is ON

{H} = {on|off} Used to set command format to hexadecimal, default value is OFF

{STRING} = Control string to transmit to 3rd party device

{A} = Alias of device

Example 1

Transmit hex characters *KA WE 4E CC* to a switcher device connected to an encoder with the alias name of *IN1* in feed back mode. The serial settings for the device is 9600bps baud rate, 8 data bits, no parity and 1 stop bit (S parameter is 9600-8n1) and add a carriage return in the end of this command.

Command

```
serial -b 9600-8n1 -r on "KA WE 4E CC" IN1
```

Response

serial command received:

```
serial -b 9600-8n1 -r on "KA WE 4E CC" IN1
```


Example 2

Transmit ASCII characters *PWR ON* to a display devices connected to decoders with the alias name of OUT1 and OUT2 in feed back mode. The serial settings for the device is 9600bps baud rate, 8 data bits, no parity and 1 stop bit (S parameter is 9600-8n1) and add a carriage return in the end of this command.

Command

```
serial -b 9600-8n1 -r off "PWR ON" IN1 IN2
```

Response

serial command received:

```
serial -b 9600-8n1 -r off "PWR ON" IN1 IN2
```

scene Commands

scene Commands Summary

Commands	Description
scene get	Queries all scene names
scene active	Enables a new scene in a video wall
scene set	Sets an encoder to a decoder in a scene of a video wall
scene change (scene - encoder)	Sets one encoder to all decoders in a scene in a video wall
scene connect (scenename)	Set encoders to corresponding decoders of a scene in sequence

scene get

Queries all scene names.

Command Structure

scene get

Example

Query all scenes for the connected IPEXCB, the current scene names are SCENE1, SCENE2 and SCENE3

Command

scene get

Response

scene list:
SCENE1 SCENE2 SCENE3

scene active

scene active enables a new scene in a video wall. This action takes effect immediately.

Command Structure

```
scene active {S}
```

{S} = Name of scene

Example

Enables a new scene, *SCENE1*, on a video wall.

Command

```
scene active SCENE1
```

Response

```
scene SCENE1 active success
```

scene set

scene set sets an encoder to a decoder in a scene of a video wall. This action makes the decoder display this source until *scene active* is executed.

Command Structure

```
scene set {S} {X} {Y} {E}
```

{S} = Name of scene

{X} = X coordinate of display location

{Y} = Y coordinate of display location

{E} = Encoder to be assigned

Example

Assign encoder *IN1* to the decoder at position 1, 2 in scene *SCENE1* of a video wall.

Command

```
scene set SCENE1 1 2 IN1
```

Response

```
Scene SCENE1 source in [1 2] change to IN1
```

scene change (scene - encoder)

scene change (scene - encoder) sets one encoder to all decoders in a scene in a video wall. This action makes the decoder display this source until `scene active` is executed.

Command Structure

scene change {S} {E}

{S} = Name of scene
{E} = Encoder to be assigned

Example

Assign a source IN1 to all decoders in *scene1* of a video wall.

Command

scene change scene1 IN1

Response

scene scene1's tx change to IN1

scene connect (scenename)

scene connect (scenename) assigns encoders to corresponding decoders of a scene in the sequence they are listed. This command only functions once and will not be saved to the IPEXCB.

Command Structure

scene connect {S} {E}

{S} = Name of scene
{E} = Encoder to be assigned

Example

Assign sources IN1 IN2 IN3 IN4 to the corresponding decoder of scene1 in sequence.

Command

scene connect scene1 IN1 IN2 IN3 IN4

Response

scene scene1's tx connect to IN1 IN2 IN3 IN4

notify Commands

notify commands are automatically sent to a third party controller connected to the IPEXCB via telnet or RS232. They show some status changes within the IP video system. A third party controller can capture this information from the session and offer it to the application layer. The commands in this section have no requests and responses.

notify Commands Summary

Commands	Description
<code>notify endpoint</code>	Queries all scene names
<code>notify serialinfo</code>	Enables a new scene in a video wall
<code>notify video {lost found}</code>	Sets one encoder to all decoders in a scene in a video wall

notify endpoint

Indicates that an encoder or decoder has come online or gone offline.

"+" indicates the device has come online. "-" indicates the device has gone offline.

Example

IPEXCB informs a third party control device that a decoder with the alias name of OUT1 just got online.

Response

notify endpoint + OUT1

notify serialinfo

notify serialinfo indicates that data has been received on a device's RS232 port.

alias is a device name which has received data. *hex* is hexadecimal format while *ascii* is ASCII format; they cannot be used in the same time. `\r` and `\n` are escape characters, meaning a carriage return and a line feed respectively.

infolen is the length of *info* in bytes. *info* is the actual data received. For ASCII data, *infolen* is the number of actual databytes received. For hexadecimal data, $(infolen+1)/3$ is the number of actual data bytes received.

Example 1

A decoder with the alias name of *OUT1* serial port receives 19 bytes which are hexadecimal characters `68 65 6C 11 6C 6F 11 22 33 44 00 55 66 77 99 AA CC DD FF`: (*infolen* is "56").

Response

```
notify serialinfo OUT1 hex 56:  
68 65 6C 11 6C 6F 11 22 33 44 00 55 66 77 99 AA CC DD FF
```

Example 2

IN1 serial port receives five characters "12345".

Response

```
notify serialinfo IN1 ascii 5:  
12345
```

notify video (5000 series only)

notify video indicates a video feed has been lost or restored.

This command only works with encoders.

Example 1

Encoder with the alias name of IN1 has lost the video stream from the connected source

Response

notify video lost IN1

Example 2

Encoder with the alias name of IN1 has restored the video stream from the connected source

Response

notify video found IN1

Digitalinx IP is a brand of:



LIBERTY[®]
AV SOLUTIONS

A SUBSIDIARY OF WESCO DISTRIBUTION, INC.

11675 Ridgeline Drive
Colorado Springs, Colorado
80921 USA
Phone: 719-260-0061
Fax: 719-260-0075
Toll-Free: 800-530-8998
Email: supportlibav@libav.com