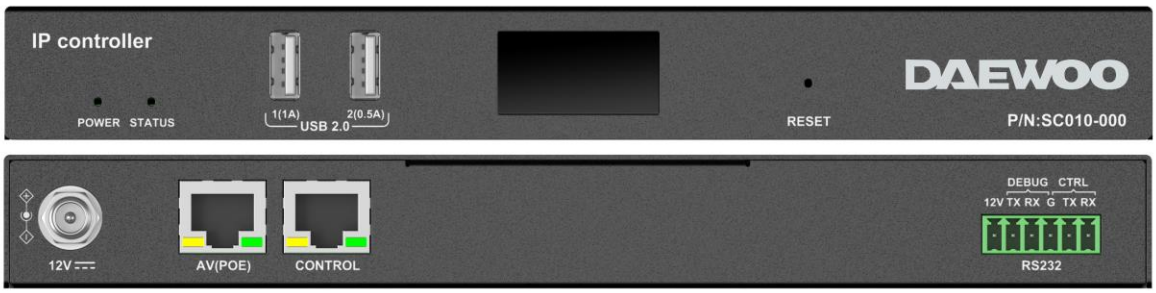
Command Guide for DSC010 Controller



Document Version: V1.0.6
Date: Jul 31st, 2024
Applicable Firmware Version: V1.0.5 or later

Contents

1. Introduction

IP controller has two Ethernet ports: LAN (AV) port and LAN(C) port. Once powered on, it will listen at TCP port 23 to try to establish a Telnet/TCP session with the remote 3rd controller on these two ports, through which you can control and manage IP matrix with the API command.

DSC010 provides SDK, with SDK use the API interface to communicate directly with IP control server. This bypasses the telnet server service, will provide Demo program for reference when SDK is available.

Important Notes

1. Alias based programming mechanism
   This mechanism allows you not necessarily to modify the 3rd controller's program when some device is damaged and has to be replaced with new one.

   To enable this, the 3rd controller should issue "config set session alias on" command once the telnet session is established and then in this session not only the command but also the feedback information is alias based (However, this rule is not suitable to the IP controller's web UI). For backward compatibility, those sessions without this mechanism enabled, the feedback information will still be based on host name.
2. Commands specified to specific models should refer to the product version.
   Sometimes you may find that this API doc claims some commands specific to some model while in fact the model doesn't support these commands. This happens most likely with the different version of that model. You can refer to the version's release notes or our post sales.
3. Gateway setting pattern
   The gateway setting pattern is single gateway, i.e. for the gateways of the LAN (AV) and LAN (C) ports, one must be set as 0.0.0.0, and the other be set as an available gateway address. The available gateway address will be the default gateway.
4. Device name replacement by keyword
   For some commands, *hostname* can be one of the keywords like *ALL_DEV*, *ALL_TX*, *ALL_RX*. When *hostname* is a keyword, the command cannot include other keywords or device names.
   Here is a list of commands that support

keywords: config set device restorefactory
config set device reboot config set
device cec standby config set
device cec onetouchplay config
set device sinkpower config set
device audio volume config set
device cec notify config get device
info
config get device
status serial
infrared
cec

1.1 Preparation

This section takes a third-party control device of windows 7 as an example. You may also use other control devices.

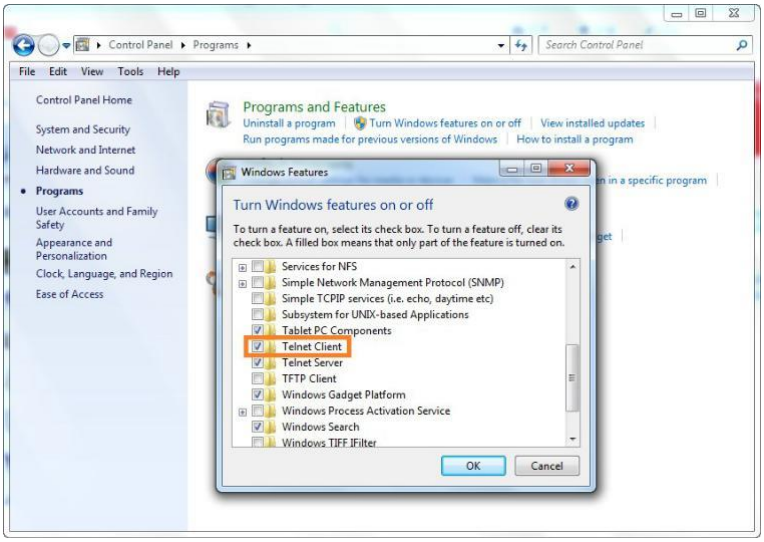
1.1.1 Setting IP Address in Your Computer

Before logging in to IP controller via command-line interface, make sure that your computer and IP controller are on the same subnet. If network settings in LAN(C) port of IP controllerare192.168.11.243/16, set your IP address in the 192.168.11.x range with a subnet mask of 255.255.0.0

1.1.2 Enabling Telnet Client

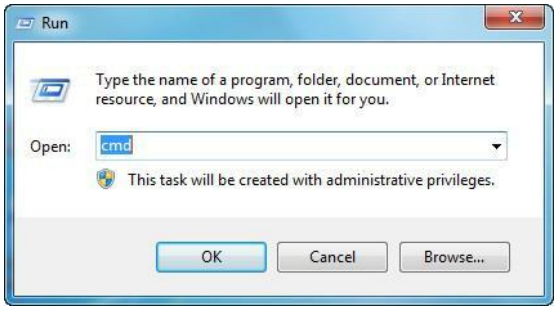
Before logging in to IP controller via command-line interface, make sure that **Telnet Client** is enabled. By default, **Telnet Client** is disabled in Windows 7. To turn on **Telnet Client**, do as follows.

1. Choose **Start** > **Control Panel** > **Programs**.
2. In **Programs and Features** area box, click **Turn Windows features on or off**.
3. In **Windows Features** dialog box, select **Telnet Client** check box.

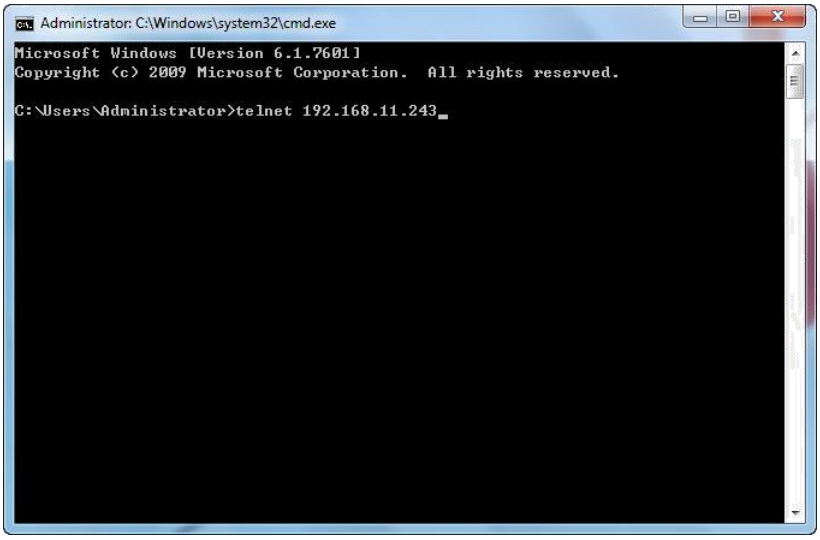


1.2 Logging In to IP Controller via Command-line Interface

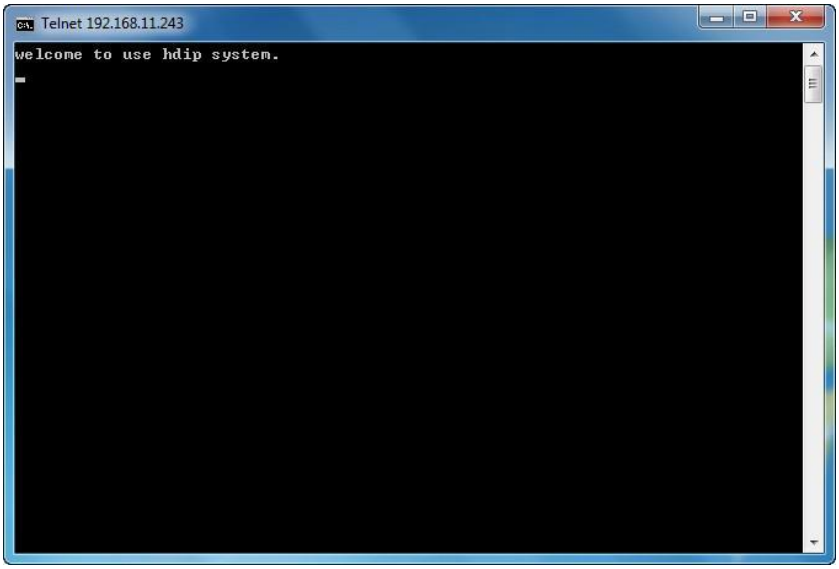
1. Choose **Start**>**Run**.
2. In the **Run** dialog box, enter **cmd** then click **OK**.



3. Enter **telnet 192.168.11.243** (IP address of DSC010 LAN(C) port), and then press **Enter**.



4. Enter API commands in the following screen to control and manage IP matrix.



1.3 Introduction to Terminology

The terminology used in API command description is listed as follows.

Terminology	Description
Device	TX, RX, a presentation switcher, a recording server controlled and managed by IP controller.
Online	Device is working properly and can be controlled by IP controller.
Offline	Device cannot be controlled by IP controller for a reason such as power failure.
Device Name	A fixed name given by factory defaults with a format "Device type-MAC address", for example DIPE5100-341B22FFFFB3.
Alias	A name given manually for easy management. It can be changed using any characters or strings except some special ones. For more information, see 2.1.8config set device alias.

1.4 API Commands Overview

API commands of IP controller are mainly classified into the following types.

- config: manages and configures IP controller and devices
- matrix: controls the switching of TX and RX or obtains matrix information

- source: obtains or selects source input in TX
- vw: configures and manages video wall
- serial: sends commands to peripheral devices via serial ports of the devices
- notify: positively informs a third party control device such as a PC about serial response and online status.

1.4.1 config Commands

config commands are mainly classified into two types **config set** and **config get** commands.

1. config set Commands

Commands	Description
config set ip4addr	Configures network settings in LAN(AV) port for communicating with devices
config set ip4addr2	Configures network settings in LAN(C) port for communicating with a third party control device such as a PC
config set telnetpasswd	Sets Telnet login password
config set delete telnetpasswd	Deletes Telnet login password
config set restorefactory	Resets IP controller to factory defaults
config set reboot	Reboots IP controller
config set device alias	Renames a device
config set device remove	Removes a device record from IP controller
config set device ip	Configures device network settings
config set device reboot	Reboots a device
config set device restorefactory	Resets a device to factory defaults
config set device info	Changes device working parameters
config set device cec standby	Makes display devices connected to RX enter standby status
config set device cec onetouchplay	Wakes up display devices connected to RX
config set device sinkpower *{on|off} hostname1 hostname2 ...*	Wakes up a display device or makes it enter its standby mode.
config set device audio input type *TYPE hostname1 hostname2 ...*	Configures device hostname1, hostname2 audio input type.
config set device status notify *{on|off} hostname*	Wakes up the device status notify system or make it enter its standby mode.
config set device cec notify *{on|off} hostname*	Wakes up the device CEC notify system or make it enter its standby mode.
config set device audio volume *{mute|unmute|up|down} {hdmi[:n]|analog[:n]|all} hostname1 hostname2 ...*	Control the device audio volume.
config set device videosource *hostname1 {auto|hdmi|dp},...*	Configure input video signal type for TX
config set device audiosource *hostname1 {hdmi|analog|dmix},...*	Configure HDMI audio source type for RX
config set device audio2source *hostname1 {analog|dmix},...*	Configure analog audio source type for RX
config set device statusseq *hostname1 1 hostname2 2 ...*	Arrange the devices of TX, RX, RX at the TRX in sequence starting from 1.
config set device statusseq4trtx *hostname1 1 hostname2 2 ...*	Arrange the devices of TX at the TRX in sequence starting from 1.

config set device group update*[json]*	Arrange specific group and devices in sequence.
config set device findme *{0-?} hostname1 hostname2 ...*	Set LED blink for specific device to locate it.
config set session alias *{on|off}*	Open or close the alias mode on current session
config set device fastswitch *{on|off} hostname1 hostname2 ...*	Turn on or off the fast switch function for certain devices.
config set telnet alias *{on|off}*	Open or close the alias mode on Telnet session
config set rs-232 alias *{on|off}*	Open or close the alias mode on rs-232 session
config set system sshservice *{on|off}*	Open or close the SSH Service of the system
config set system workmode	Configure working mode for the system
config set system preview fps *{FPS}*	Configure the preview framerate for the IP6000 TX series in the system
config set system auth *{on|off}*	Configure security settings for the IP6000 series in the system
config set system apireplace	Configure information of source API and target API
config set system ldap	Configure LDAP for the controller
config set system 802_1x	Configure 802.1x for the controller
config set system lcd {*ipversion*|*logo*|*off*}	Configure the LCD display content of the controller.
config set dnsserver ip4addr xx.xx.xx.xx	Configure the preferred DNS server's IP address for the controller's network port.
config set dnsserver ip4addr2 xx.xx.xx.xx	Configure the alternate DNS server's IP address for the controller's network port.
config set system xyte_setting	Initiate service registration request for specified devices to the cloud server of the xyte platform.
config set system default xyte_setting	Restore the controller's registered URL of cloud service to factory defaults.

Note:

"config set device info" does not apply to DIPX-5100 series products.

2. config get Commands

Commands	**Description**
config get version	Obtains IP controller version information
config get devicelist	Obtains an online device list
config get ipsetting	Obtains network settings in LAN(AV) port
config get ipsetting2	Obtains network settings in LAN(C) port
config get name	Obtains a device name or its alias
config get device info	Obtains device working parameters
Config get device status	Obtains device status information
config get devicejsonstring	Obtains all device information
config get scenejsonstring	Obtains all scene information
config get telnet alias	Obtains the alias mode on Telnet session
config get rs-232 alias	Obtains the alias mode on rs-232 session
config get system sshservice	Obtains the SSH Service of the system
config get system info	Obtains the system status
config get system apireplace	Obtains the information of source API and target API
config get system ldap	Obtains the LDAP configuration information of the controller
config get system 802_1x	Obtains the 802.1x configuration information of the controller

config get system lcd	Obtains the LCD display's configuration information of the controller.
config get controller info	Obtain the controller's information, including hostname, mac address, serial no., version information, etc.
config get dnsserver ip4addr	Obtain the preferred DNS server's IP address for the controller's network port.
config get dnsserver ip4addr2	Obtain the alternate DNS server's IP address for the controller's network port.

1.4.2 matrix Commands

Command	Description
matrix set	Controls switching of TX and RX
matrix get	Obtains TX played by RX in matrix
matrix video set TX1 RX1 RX2, TX2 RX3 RX4, ...	Change the RX and TX video matrix link relationship
matrix video get RX1 RX2 …	Get all or parts of the RX information which link relationship with TX
matrix audio set TX1 RX1 RX2, TX2 RX3 RX4, ...	Change the RX and TX matrix link relationship in the audio matrix
matrix audio get RX1 RX2 …	Get all or parts of the RX information which link relationship with TX in audio matrix
matrix usb set TX1 RX1 RX2, TX2 RX3 RX4,..	Change the USB matrix link relationship
matrix usb get RX1 RX2 …	Get all or parts of the RX information which link relationship with TX in USB matrix
matrix audio2 set TX1 RX1 RX2,TX2 RX3 RX4,...	Change the RX and TX analog audio matrix link relationship
matrix audio2 get RX1 RX2	Get all or parts of the RX link information with TX in analog audio matrix
matrix audio3 set *TX1 RX1 RX2,TX2 RX3 RX4,...*	Set the link relationship between RX and TX in the ARC audio matrix.
matrix audio3 get *TX1 TX2*	Get which RX is linked to the specified TX(s) in the ARC audio matrix.
matrix infrared set TX1 RX1 RX2, TX2 RX3 RX4, ...	Change the RX and TX infrared matrix link relationship
matrix infrared get RX1 RX2 …	Get all or parts of the RX link information with TX in infrared matrix
matrix serial set TX1 RX1 RX2, TX2 RX3 RX4, ...	Change the RX and TX serial matrix link relationship
matrix serial get RX1 RX2 ...	Get all or parts of the RX link information with TX in serial matrix

1.4.3 scene Commands

Command	Description
scene get	Obtains all scene names
scene active *scenename*	Enables a new scene in video wall. This action takes effect immediately.
scene set *scenename posX posY tx1…*	Assigns a source to RX in a scene of video wall. This action makes RX display this source until **scene active *scenename*** is executed.
scene change *scenename txname*	Assigns a source to all RX in a scene of video wall. This action makes all RX display this source until **scene active *scenename*** is executed.
scene set *sceneName* bezelgap *vw-nameow oh vw vh*	Sets the bezel compensation parameters for a specific video wall in a scene.
scene set *sceneName* stretch *vw-name type*	Sets the stretch mode for a specific video wall in a scene.
scene connect *scenename tx1 tx2 … txnm*	Assigns sources to the corresponding RX of a scene in sequence. This action is operated only once and will not be saved in IP controller.
scene create *scenesjsonstring*	Create new scene(s) that defined in json string.

scene update *scenesjsonstring*	Use the configuration information that defined in json string to update the specified scene.
scene modify name *scenename_old1 scenename_new1 scenename_old2 scenename_new2 ...*	Change the names of specified scenes to new ones.
scene remove *scenename1 scenename2 ..*	Remove the specified scene(s).
scene group update *[json]*	Update the group information of the scenes.

1.4.4 vw Commands

Command	Description
vw add	Creates video wall
vw rm	Removes video wall
vw rmvwname rx	Removes one or multiple RX from video wall
vw add position	Adds RX to video wall
vw add layout	Creates video wall and automatically applies the settings
vw change rx tx	Removes a certain RX from video wall
vw change vw-name tx	Changes to another source for video wall
vw bezelgap	Sets bezel compensation parameters
vw get	Obtains a list of all video walls
vw stretch vw-name type	Sets the stretch mode of the video wall

Note:

"vw bezelgap" applies to JPEG2000 series products but does not apply to H.264 series products.

1.4.5 serial Commands

Command	Description
serial	Sends commands to peripheral devices via serial ports of the devices

1.4.6 “command” Commands

Command	Description
command device-name message-body	Sends shell commands to the corresponding TX/RX device via the control box

1.4.7 notify Commands

Command	Description
notify endpoint	IP controller positively informs third party control device that devices just got online or offline when devices' online or offline status changes.
notify serialinfo	IP controller positively informs third party control device such as a computer about the data received in a device's serial port.

notify irinfo *dev "IRDATA"*	IP controller positively informs third party control device such as a computer about the IR data of a device.
notify cecinfo *dev* "*CECDATA*"	IP controller positively informs third party control device such as a computer about the cec information of a device.
notify video {lost|found} tx/rx tx/rx	IP controller positively informs third party control device or application that some TX/RX lost the video signal or restored.
notify device status	Notify device status, the devices' status is described by json format.
notify update*DEV status*	The controller proactively informs third party control device such as a computer about the update status of a device.

1.4.8 mscene Commands

Command	Description
mscene get	Obtain all mscene name
mscene getjson mrx1	Obtain a specify mscene json information
mscene active mrxname layoutname	Apply the specify layout in the specify MRX
mscene change mrxname layoutname window1 txname1	Change the source (tx) for certain windows in the specify layout of the specify MRX
mscene changeall mrxname layoutname txname1	Change the source (tx) for all windows in the specify layout of the specified MRX
mscene set audio mrxname layoutname *{follow|window|separate} [{windowname|tx}]*	Configure the audio mode for the specify layout of the specify MRX
mscene changeaudio mrxname *{follow|separate} [tx]*	Change the audio mode for all layouts of the specify MRX
mscene create *[]*	Create json descriptive information for all scenes or specified scenes.
mscene group update *[json]*	Change grouping or sequence.
mscene set source mrxname la*youtnam*e	Set the source of the screen for specified scene, which takes effect in active/change scene.
mscene set pipposition mrxname *layoutname {0/1/2/3}*	Set the position of the small screen in pip scene.
mscene set pipsize mrxname *layoutname {0/1/2}*	Set the aspect ratio of the small screen size in pip scene.

1.4.9 wscene2 Commands

Command	Description
wscene2 get	Obtain all names of windowing scenes.
wscene2 getjson *wscenename*	Obtain descriptive information of all or specified windowing scenes in json string.
wscene2 active *wscenename*	Apply the specified windowing scene.
wscene2 create *wscenesjsonstring*	Create a windowing scene that defined by *wscenesjsonstring* in server.
wscene2 update wscene2sjsonstring	Apply described scene information in wscene2sjsonstring and save them in server
wscene2 modify name *scenename_old scenename_new*	Change the name of a windowing scene to a new one.

Command	Description
wscene2 remove *scenename1 scenename2 ..*	Remove the specified scene(s).
wscene2 group update *[json]*	Update the group information of the scenes.
wscene2 window open *scenename windowname hstart vstart hsize vsize txname*	Create a window in the specified scene.
wscene2 window close *scenename windowname*	Close the specified window in the corresponding windowing scene.
wscene2 window adjust *scenename windowname hstart vstart hsize vsize*	Adjust the specified window's position and size.
wscene2 window move *scenename windowname {up|down|top|bottom}*	Arrange the layer order for the specified window.
wscene2 window change *scenename windowname1 txname1*	Change the source (tx) for the specified windows in the corresponding windowing scene.
wscene2 window changeall *scenename txname*	Change the source for all windows in the corresponding windowing scene.
wscene2 window changetx *scenename windowname1 txname1 windowname2 txname2…*	Assign a source to a window one by one in the corresponding windowing scene.

1.4.10 Infrared Commands

Command	Description
infrared IRDATA hostname	Transmits IR commands to the device

1.4.11 cec Commands

Command	Description
cec *CECDATA hostname*	Transmits CEC commands to the device

1.4.12 update Commands

Command	Description
update *firmwarename dev1…*	Update the firmware.

1.4.13 canvas Commands

Command	Description
canvas get *canvasname*	Get the layout name of canvasname
canvas getjson *canvasname*	Get the json information of the specified canvas
canvas layout getjson *canvasname layoutname*	Get one layout's json information of the specified canvas
canvas active *canvasname layoutname*	Apply one specified layout for the video wall
canvas changetx *canvasname layoutname txname*	Change all windows' video sources in a layout for the specified video wall, and apply this layout
canvas window changetx *canvasname layoutname windowname txname*	Configure the specified window within the specified layout on the canvas to play the "*txname*" video source
canvas group update *[json]*	Arrange the group and the mosaic style video wall in sequence.
canvas screenproperties getjson *brand model*	Obtain the attribute information in json format of the TVs that make up the mosaic style video wall.

Command	Description
canvas screenproperties create *canvasjsonstring*	Create or update the attribute information in json format of the TVs that make up the mosaic style video wall.
canvas screenproperties remove *brand1 model1*	Remove the TV screen, including brand and model no.
canvas create *[canvasjsonstring]*	Update the layout that exists in controller using the json information
canvas modify name *canvasname_old canvasname_new*	Modify the name of the mosaic style video wall.
canvas remove *canvasname1 canvasname2 ..*	Remove the specified scene names.

1.4.14 mrscene Commands

Command	Description
mrscene get *[mrscene1]*	Obtains the layout names of the specified mouse roaming scenes.
mrscene getjson *[mrscene1]*	Obtains the information in json string of the specified mouse roaming scene.
mrscene active *mrscene1*	Apply mouse roaming for the specified layout on the video wall.
mrscene create *[mrscenejsonstring]*	Create a mouse roaming scene that defined by *[mrscenejsonstring]* in server and apply it to the specified device.
mrscene update *[mrscenejsonstring]*	Use the configuration information that defined by *[mrscenejsonstring]* to update a mouse roaming scene in server and apply it to the specified device.
mrscene modify name *scenename_old scenename_new*	Change the name of a mouse roaming scene to a new one.
mrscene remove *scenename1 scenename2 ..*	Remove the specified mouse roaming scene(s).
mrscene group update *[json]*	Update the group information of mouse roaming scenes.

2. Command Sets

2.1 config Commands

2.1.1 config set ip4addr

Command	`config set ip4addr` *`xx.xx.xx.xx`* `netmask` *`xx.xx.xx.xx`* `gateway` *`xx.xx.xx.xx`*
Response	`ip setting will change to: ipaddr` *`xx.xx.xx.xx`* `netmask` *`xx.xx.xx.xx`* `gateway` *`xx.xx.xx.xx`*
Description	Configures network settings in LAN(AV) port for communicating with devices **Note:** • This command is used to set IP address, subnet mask and gateway in LAN(AV) port. You can set two or three of them at the same time or only one each time. • LAN(AV) port only supports Static IP mode. After network settings are configured, it automatically reboots for the settings to take effect.

Example:

If you want to set LAN(AV) port's IP address as 169.254.1.254, subnet mask 255.255.0.0 and gateway 169.254.1.1:

Command:

```
config set ip4addr 192.168.1.100 netmask 255.255.0.0 gateway 192.168.1.1
```

Response:

```
ip setting will change to: ipaddr 192.168.1.100 netmask 255.255.0.0 gateway
192.168.1.1
```

2.1.2 config set ip4addr2

Command	config set ip4addr2 *xx.xx.xx.xx* netmask *xx.xx.xx.xx* gateway *xx.xx.xx.xx*
Response	ip2 setting will change to: ipaddr *xx.xx.xx.xx* netmask *xx.xx.xx.xx* gateway *xx.xx.xx.xx*
Description	Configures network settings in LAN(C) port for communicating with a third-party control device such as a PC. **Note:** • This command is used to set IP address, subnet mask and gateway in LAN(C) port. You can set two or three of them at the same time or only one each time. • LAN(C) port only supports Static IP mode. After network settings are configured, it automatically reboots for the settings to take effect.

Example:

If you want to set LAN(C) port's IP address as 192.168.11.243, subnet mask 255.255.0.0 and gateway 192.168.11.1:

Command:

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

Response:

```
ip2 setting will change to: ipaddr 192.168.11.243 netmask 255.255.0.0 gateway
192.168.11.1
```

2.1.3 config set telnetpasswd

Command	config set telnetpasswd *xxxxxx*
Response	password for telnet modified
Description	Sets Telnet configuration page login password. Please use the new one for next login.

Example:

If you want to change login password to 123456:

Command:

```
config set telnetpasswd 123456
```

Response:

```
password for telnet modified
```

2.1.4 config set delete telnetpasswd

Command	`config set delete telnetpasswd`
Response	`password for telnetdeleted`
Description	Delete Telnet configuration page login password. Please use the new one for next login.

Example:

If you want to delete login password:

Command:

```
config set delete telnetpasswd
```

Response:

```
password for telnet deleted
```

2.1.5 config set restorefactory

Command	`config set restorefactory`
Response	`system will restore to factory settings now`
Description	Resets IP controller to factory defaults. When it is restored to factory defaults, it will automatically reboot for the settings to take effect.

Example:

If you want to reset IP controller to factory defaults:

Command:

```
config set restorefactory
```

Response:

```
system will restore to factory settings now
```

2.1.6 config set reboot

Command	`config set reboot`
Response	`system will reboot now`
Description	Reboots IP controller

Example:

If you want to reboot IP controller:

Command:

```
config set reboot
```

Response:

```
system will reboot now
```

2.1.7 config set device alias

Command	config set device alias *hostnamexxxx*
Response	hostname’s alias is *xxxx*
Description	Renames device

Note:

- **hostname** is device name.
- Alias can be used in other commands to replace its device name.
- Alias should be different from others.
- Alias cannot contain the characters (exclude the double quotation marks) in the following table. "NULL" is not case sensitive.

" " (space)	","	";"	"_"	"@"	"*"
"&"	"NULL"				

Example:

If you want to setDIPD-5100-341B22FFFFB3's alias as MYDVD:

Command:

```
config set device alias DIPD5100-341B22FFFFB3 MYDISPLAY
```

Response:

```
DIPD5100-341B22FFFFB3's alias is MYDISPLAY
```

2.1.8 config set device remove

Command	config set device remove *hostname1 hostname2*...
Response	The following device’s record will be removed: *hostname1* *hostname2* …
Description	Removes a device record from IP controller. **Note:** • **hostname1**and **hostname2** are device names. • You can remove one or multiple devices' records at one time. When a device's record is removed, it cannot be detected and controlled by IP controller. If you want to restore the removed online device, reboot it or IP controller. If you want to restore the removed offline device, reboot it.

Example:

If you want to remove the records ofDIPD-5100-AABBCCEEDDFF andDIPD-5100-1234567890AB:

Command:

```
config set device remove DIPD-5100-AABBCCEEDDFFDIPD-5100-1234567890AB
```

Response:

```
the following device's record will be removed:
```

IPD5100-AABBCCEEDDFF

IPD5100-1234567890AB

2.1.9 config set device ip

Command	config set device ip *hostname1 {autoip|dhcp|static ip4addr netmask gateway}, hostname2 {autoip|dhcp|static ip4addr netmask gateway} ...*
Response	Devices' ipsetting will change to: *hostname1 {autoip|dhcp|static ip4addr netmask gateway}* *hostname2 {autoip|dhcp|static ip4addr netmask gateway}* …
Description	Configures device network settings. **Note:** • **hostname1**and **hostname2** are device names. • Devices support AutoIP, DHCP and Static IP for network configuration. For Static IP, you need to set IP address, subnet mask and gateway at the same time. • You can use configure network settings for multiple devices at one time. • After network settings are configured, you must reboot the devices for the settings to take effect. This command will not restart devices.

Example:

If you want to setDIPD-5100-341B22800BCD to AutoIP and IPD500-341B22800BCAto Static IP (IP address 169.254.5.253, subnet mask 255.255.0.0, gateway 169.254.1.253):

Command:

```
config set device ip DIPD-5100-341B22800BCD autoip,DIPD-5100-341B22800BCA static
169.254.5.253 255.255.0.0 169.254.1.253
```

Response:

```
Devices's ipsetting will change to:
    IPD5100-341B22800BCD autoip
IPD5100-341B22800BCA static 169.254.5.253 255.255.0.0 169.254.1.253
```

2.1.10 config set device reboot

Command	config set device reboot *hostname1 hostname2...*
Response	the following device will reboot now: *hostname1* *hostname2* *...*

Description	Reboots one or multiple devices. **Note:** **hostname1** and **hostname2** are device names.

Example:

If you want to rebootDIPD-5100-341B22FFFFB3 andDIPD-5100-341B22FFFFB4:

Command:

```
config set device reboot DIPD-5100-341B22FFFFB3DIPD-5100-341B22FFFFB4
```

Response:

```
the following device will reboot now:
IPD5100-341B22FFFFB3
IPD5100-341B22FFFFB4
```

2.1.11 config set device restorefactory

Command	`config set device restore factory` *`hostname1 hostname2...`*
Response	`the following device will restore to factory setting now:` *`hostname1`* *`hostname2`* *`...`*
Description	Resets one or multiple devices to factory defaults. After they are restored to factory defaults, devices will automatically reboot for the settings to take effect. **Note:** **hostname1** and **hostname2** are device names.

Example:

If you want to reset DIPE5100-341B22FFFFB3 and DIPE5100-341B22FFFFB4 to factory defaults:

Command:

```
config set device restorefactory DIPE5100-341B22FFFFB3 DIPE5100-341B22FFFFB4
```

Response:

```
the following device will restore to factory setting now:
DIPE5100-341B22FFFFB3
DIPE5100-341B22FFFFB4
```

2.1.12 config set device info

Command	`config set device info` *`key1=value1 [key2=value2…] hostname1 hostname2...`*
Response	`config set device info` *`key1=value1 key2=value2 key3=value3`*

	key4=value4 hostname1 hostname2...
Description	Changes a device's one or multiple working parameters in **key=value** format. You can change parameters for multiple devices at one time. **Note:** • **hostname1** and **hostname2** are device names. • **Key** is parameter name and **value** is its value. For more information, see 3.1Device Info section.

Example:

If you want to set DIPD-5100-AABBCCDDEEFF's **mic_volume** as 20, **audio.mic1.gain** 12 and **audio.lineout1.volume**20:

Command:

```
config set device info mic_volume=20 audio.mic1.gain=12
audio.lineout1.volume=20DIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device info mic_volume=20 audio.mic1.gain=12
audio.lineout1.volume=20DIPD-5100-AABBCCDDEEFF
```

2.1.13 config set device cec standby

Command	`config set device cec standby` *hostname1 hostname2...*
Response	`config set device cec standby` *hostname1 hostname2...*
Description	Makes one or multiple display devices connected to RX enter standby status. **Note:** • **hostname1** and **hostname2** are device names. • This command is used to control RX to send a CEC command to make one or multiple display devices enter standby mode. • You can just use one command to make multiple display devices enter standby mode. • Display devices must support CEC.

Example:

If you want a display device connected to RXDIPD-5100-AABBCCDDEEFF enter standby mode:

Command:

```
config set device cec standbyDIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device cec standbyDIPD-5100-AABBCCDDEEFF
```

2.1.14 config set device cec onetouchplay

Command	`config set device cec onetouchplay`*hostname1 hostname2...*
Response	`config set device cec onetouchplay` *hostname1 hostname2...*
Description	Wakes up one or multiple display devices connected to RX.

	Note: • **hostname1** and **hostname2** are device names. • This command is used to control RX to send a CEC command to wake up one or multiple display devices. • You can just use one command to wake up multiple display devices. • Display devices must support CEC.

Example:

If you want to wake up a display device connected to RXDIPD-5100-AABBCCDDEEFF:

Command:

```
config set device cec onetouchplayDIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device cec onetouchplayDIPD-5100-AABBCCDDEEFF
```

2.1.15 config set device sinkpower

Command	`config set device sinkpower {on|off} hostname1 hostname2 ...`
Response	`config set device sinkpower {on|off} hostname1 hostname2 ...`
Description	Wakes up a display device or makes it enter its standby mode.

Example:

If you want to wake up a display device connected toDIPD-5100-AABBCCDDEEFF from its standby mode:

Command:

```
config set device sinkpower onDIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device sinkpower onDIPD-5100-AABBCCDDEEFF
```

2.1.16 config set device audio

Command	`config set device audio input type TYPE hostname1 hostname2 ...`
Response	`config set device audio input type TYPE hostname1 hostname2 ...`
Description	This command is only used for DIPE5100, configure device hostname1, hostname2's audio input type such as auto, hdmi, analog.

Example:

Command:

```
config set device audio input type hdmi DIPE5100-AABBCCDDEEFF
```

Response:

```
config set device audio input type hdmi DIPE5100-AABBCCDDEEFF
```

2.1.17 config set device status notify

Command	`config set device status notify{on|off} hostname1 hostname2 ...`
Response	`config set device status notify {on|off} hostname1 hostname2 ...`
Description	Wakes up device status notify or makes it enter its standby mode. hostname is the device alias; Hostname also can be KEY words: ALL_DEV, ALL_TX, ALL_RX when hostname is one of the KEY word, this command will not include other KEY word and device name. **Note:** • This command is available for DIPX-5100. • When the system work mode is set as 1, this command will be not available.

Example 1:

Command:

```
config set device status notify on DIPE5100-AABBCCDDEEFF
```

Response:

```
config set device status notify on DIPE5100-AABBCCDDEEFF
```

Example 2:

Command:

```
config set device status notify on ALL_TX
```

Response:

```
config set device status notify on ALL_TX
```

2.1.18 config set device cec notify

Command	`config set device cec notify{on|off} hostname1 hostname2 ...`
Response	`config set device cec notify {on|off} hostname1 hostname2 ...`
Description	Wakes up device cec notify system or makes it enter its standby mode. hostname is the device alias; Hostname also can be KEY words: ALL_DEV, ALL_TX, ALL_RX, when hostname is one of the KEY word, this command will not include other KEY word and device name.

Example 1:

Command:

```
config set device cec notify on DIPE5100-AABBCCDDEEFF
```

Response:

```
config set device cec notify on DIPE5100-AABBCCDDEEFF
```

Example 2:

Command:

```
config set device cec notify on ALL_DEV
```

Response:

```
config set device cec notify on ALL_DEV
```

2.1.19 config set device audio volume

Command	config set device audio volume *{mute|unmute|up|down|digit} {hdmi[:n]|analog[:n]|all}hostname1 hostname2 ...*
Response	config set device audio volume *{mute|unmute|up|down} {hdmi[:n]|analog[:n]|all} hostname1 hostname2 ...*
Description	Control device audio volume, the meanings of parameters as follow: {mute|unmute|up|down}: up is volume increased; down is volume decreased; mute means mute mode, unmute means mute mode cancelled; {hdmi[:n]|analog[:n]|all}: hdmi means that all the HDMI audio outputs, hdmi[:n] means that the number of hdmi audio output is n; analog means that all the analog audio outputs, analog[:n]means that the number of analog audio output is n; all is all of the hdmi and analog audio outputs. **Note:** DIPX-5100 supports "up" and "down" setting for analog audio only.

Example:

If you want to increase all the analog outputs audio volume ofDIPD-5100-1 andDIPD-5100-2:

Command:

```
config set device audio volume up analogDIPD-5100-1DIPD-5100-2
```

Response:

```
config set device audio volume up analogDIPD-5100-1DIPD-5100-2
```

2.1.20 config set device videosource (for IPX6000)

Command	config set device videosource *hostname1 {auto|hdmi|dp},...*
Response	config set device videosource *hostname1 {auto|hdmi|dp},...*
Description	Configure input video signal type for TX; the signal type supports three modes: auto, hdmi and dp. **Note**: This command is available for IPX6000 only.

Example:

If you want to set input video signal type of one TX as “auto”:

Command:

```
config set device videosource TX1 auto
```

Response:

```
config set device videosource TX1 auto
```

2.1.21 config set device audiosource (only for IPX6000)

Command	config set device audiosource *hostname1 {hdmi|analog|dmix},...*
Response	config set device audiosource *hostname1 {hdmi|analog|dmix},...*
Description	Configure HDMI audio source type for RX; the source type supports three modes: hdmi (digital audio, corresponds to audio from DP input or HDMI input), analog and dmix.

	Note: This command is available for IPX6000 only.

Example:

If you want to set HDMI audio source type of one RX as “hdmi”:

Command:

```
config set device audiosource RX1 hdmi
```

Response:

```
config set device audiosource RX1 hdmi
```

2.1.22 config set device audio2source (only for IPX6000)

Command	`config set device audio2source` *`hostname1 {analog|dmix},...`*
Response	`config set device audio2source` *`hostname1 {analog|dmix},...`*
Description	Configure audio source type for RX’s analog output port; the source type supports two modes: analog and dmix. **Note**: This command is available for IPX6000 only.

Example:

If you want to set audio source type for one RX’s analog output port as “analog”:

Command:

```
config set device audio2source RX1 analog
```

Response:

```
config set device audio2source RX1 analog
```

2.1.23 config set device statusseq

Command	`config set device statusseq` *`hostname1 1 hostname2 2 ...`*
Response	`config set device statusseq` *`hostname1 1 hostname2 2 ...`*
Description	Arrange the devices of *hostname1, hostname2, … hostnameN* in sequence starting from 1. If sequence is not set, it will start from 0 by default. Note: This command is used to arrange TX, RX, RX at the TRX in sequence.

Example:

If you want to arrange specific devices in sequence starting from 1:

Command:

```
config set device statusseq DIPE5100-E4CE0211EEB6 1 DIPE5100-E4CE02102F80 2
DIPE5100-E4CE02102F7C 3 DIPE5100-341B22000010 4 DIPE5100-361B22094005 5 DIPE5100-
341B22801F0C 6 DIPE5100-E4CE02123321 7 DIPE5100-341B22F001C0 8 DIPE5100-
341B22000009 9 DIPE5100-341B228106B8 10
```

Response:

```
config set device statusseq DIPE5100-E4CE0211EEB6 1 DIPE5100-E4CE02102F80 2 DIPE5100-
E4CE02102F7C 3 DIPE5100-341B22000010 4 DIPE5100-361B22094005 5 DIPE5100-341B22801F0C 6
DIPE5100-E4CE02123321 7 DIPE5100-341B22F001C0 8 DIPE5100-341B22000009 9 DIPE5100-
```

```
341B228106B8 10
```

2.1.24 config set device statusseq4trtx

Command	`config set device statusseq4trtx` *`hostname1 1 hostname2 2 ...`*
Response	`config set device statusseq4trtx` *`hostname1 1 hostname2 2 ...`*
Description	Arrange the devices of *hostname1, hostname2, … hostnameN* in sequence starting from 1. If sequence is not set, it will start from 0 by default. Note: This command is used to arrange TX at the TRX in sequence.

Example:

If you want to arrange specific TX of TRX devices in sequence starting from 1:

Command:

```
onfig set device statusseq4trtx DIPE5100-E4CE0211EEB6 1 DIPE5100-E4CE02102F80 2
DIPE5100-E4CE02102F7C 3 DIPE5100-341B22000010 4 DIPE5100-361B22094005 5 DIPE5100-
341B22801F0C 6 DIPE5100-E4CE02123321 7 DIPE5100-341B22F001C0 8 DIPE5100-
341B22000009 9 DIPE5100-341B228106B8 10
```

Response:

```
config set device statusseq4trtx DIPE5100-E4CE0211EEB6 1 DIPE5100-E4CE02102F80 2
DIPE5100-E4CE02102F7C 3 DIPE5100-341B22000010 4 DIPE5100-361B22094005 5 DIPE5100-
341B22801F0C 6 DIPE5100-E4CE02123321 7 DIPE5100-341B22F001C0 8 DIPE5100-
341B22000009 9 DIPE5100-341B228106B8 10
```

2.1.25 config set device group update

Command	`config set device group update` *`[json]`*
Response	`config set device group or sort update`
Description	Arrange specific group and devices in sequence. Note: • To arrange each TX and RX in a group in sequence, use “sequence”; • To arrange RX at the TRX in sequence, use the field “sequence”, for TX at the TRX, use the field “sequence4tr”.

Example:

If you want to sequence devices in a certain group:

Command:

```
config set device group update
[{"group":[{"name":"group1","sequence":0}],"sequence":0,"sequence4tr":0,"name":"I
PE5100-112233445566"}]
```

Response:

```
device group or sort update success
```

2.1.26 config set device findme

Command	`config set device findme {0-?} hostname1 hostname2 ...`
Response	`config set device findme {0-?} hostname1 hostname2 ...`
Description	Configure LED blink for specific device to locate it. ”0” represents LED blinking stops; other integers represent the duration of the LED blinking. Each LED blinks four times per second.

Example:

If you want to make the LED blink for a specific device:

Command:

```
config set device findme 60DIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device findme 60DIPD-5100-AABBCCDDEEFF
```

2.1.27 config set device fastswitch (for DIPX-5100)

Command	`config set device fastswitch {on|off} hostname1 hostname2 ...`
Response	`config set device fastswitch {on|off} hostname1 hostname2 ...`
Description	• For TX: To Turn on/off the fast switch function for specified TX devices. • For RX: To turn on the fast switch function for specified RX devices only. Note: Once you turn on the fast switch function for a certain RX, you cannot turn off this function for the RX any longer until you reset it.

Example:

Command:

```
config set device fastswitch onDIPD-5100-AABBCCDDEEFF
```

Response:

```
config set device fastswitch onDIPD-5100-AABBCCDDEEFF
```

2.1.28 config set session alias

Command	`config set session alias {on|off}`
Response	`config set session alias {on|off}`
Description	Open or close the alias mode on the current session, if the value set to be on, then all API command next to it will get alias information feedback, while the feedback got alias. If the value set to be off, then all API command next to it will get true name information feedback.

Example:

If you want to configure the session alias to ON mode:

Command:

```
config set session alias on
```

Response:

```
config set session alias on
```

2.1.29 config set telnet alias

Command	`config set telnet alias {on|off}`
Response	`config set telnet alias {on|off}`
Description	Configure the Telnet session default alias mode, it will not affect the telnet session that has been linked, only affect the telnet session which is linked later. When the value is on, the API response will describe the device with alias. When the value is off, the API response will describe the device with true name. Note: on is by default.

Example:

If you want to configure the telnet alias to off mode:

Command:

```
config set telnet alias off
```

Response:

```
config set telnet alias off
```

2.1.30 config set rs-232 alias

Command	`config set rs-232 alias {on|off}`
Response	`config set rs-232 alias {on|off}`
Description	Configure uart session alias mode. When it is on, the API response will describe the device with alias, when is off, API response will describe the device with true name. Note: on is by default.

Example:

If you want to configure the uart alias to off mode:

Command:

```
config set rs-232 alias off
```

Response:

```
config set rs-232 alias off
```

2.1.31 config set system sshservice

Command	`config set systemsshservice {on|off}`
Response	`config set system sshservice {on|off}`
Description	Open or close the system SSH service, off is by default.

Example:

If you want to open the system SSH service mode:

Command:

```
config set system sshservice on
```

Response:

```
config set system sshservice on
```

2.1.32 config set system workmode

Command	`config set system workmode {0|1}`
Response	`config set system workmode {0|1}`
Description	Set the working mode for the system. By default, it is set as mode 1. • 0: In this mode, all the IP series units except IPX6000 are available for API control. • 1: In this mode, IPX6000 is available for API control, while the other IP series units will be unavailable for API "Notify devices status". **Note:** • Please reboot the unit for this command setting to take effect.

Example:

If you want to set the system's working mode as 1:

Command:

```
config set system workmode 1
```

Response:

```
config set system workmode 1
```

2.1.33 config set system preview fps (for DIPE6000)

Command	`config set system preview fps FPS`
Response	`config set system preview fps FPS`
Description	Set the total preview framerate for the IP6000 series TX units in the system; the range is [0,30], and the type is integer. By default, the framerate is set as 0. • 0: The preview function is disabled. • Other value: The preview function for IP6000 TX is enabled; the preview framerate for each TX is calculated by system (=total framerate/quantity of online TX), the minimum framerate is 0.5.

Example:

If you want to set the total framerate for the IP6000 series TX as 30 in the system:

Command:

```
config set system preview fps 30
```

Response:

```
config set system preview fps 30
```

2.1.34 config set system auth (for IP6000)

Command	`config set system auth {on|off}`
Response	`config set system auth {on|off}`
Description	For security settings on IP6000 series units. • on: Encryption settings are enabled for IP6000 units. • off: Encryption settings are disabled for IP6000 units. By default, it is set as off.

Example:

If you want to disable security setting for IP6000 series units in the system:

Command:

```
config set system auth off
```

Response:

```
config set system auth off
```

2.1.35 config set system ldap

Command	config set system ldap
Response	config set system ldap success or UNSUPPORTED_FEATURE
Description	Configures LDAP settings. **Note:** If the specified API is not supported by the IP controller, it will return "UNSUPPORTED_FEATURE".

Example:

If you want to configure LDAP for the IP controller:

Command:

```
config set system ldap:
[
    {
     "ldap_enable": "true/false",
     "ldap_uri": "ldaps://192.168.17.225:636",
    "ldap_mode":"dn/uid"
"ldap_uid": " jeremy",
     "ldap_base_dn": "dc=my-domain, dc=com",
     "ldap_bind_dn": "cn=jeremy, ou=People, dc=my-domain, dc=com",
     "ldap_attr": "uid",
     "ldap_password": "password"
    }
]
```

Response:

```
config set system ldap success
```

2.1.36 config set system 802_1x

Command	config set system 802_1x

Response	config set system 802_1x success or UNSUPPORTED_FEATURE
Description	Configures 802.1x settings. **Note:** If the specified API is not supported by the controller, it will return "UNSUPPORTED_FEATURE".

Example:

If you want to configure 802.1x for the IP controller:

Command:

```
config set system 802_1x
[
    {
        "ieee802_1x_enable": "true",
       "ieee802_1x_mode": "mschapv2/tls"
        "ieee802_1x_mschapv2_user": "user",
"ieee802_1x_mschapv2_password": " password",
        "ieee802_1x_tls_user": "user",
       "ieee802_1x_tls_private_key_password":"password";
       "ieee802_1x_ca_mode":"true" // Authentication status of CA certificate
    }
]
```

Response:

```
config set system 802_1x success
```

2.1.37 config set system lcd {ipversion|logo|off}

Command	config set system lcd {ipversion|logo|off}
Response	config set system lcd {ipversion|logo|off}
Description	Configure the LCD display content of the IP controller. • ipversion: Displays IP address and API version. • logo: Displays logo. • off: Turns off the LCD display. By default, it displays IP address and API version.

Example:

If you want LCD to display IP address and API version for the IP controller:

Command:

```
config set system lcd ipversion
```

Response:

```
config set system lcd ipversion
```

2.1.38 config set dnsserver ip4addr

Command	`config set dnsserver ip4addr` *xx.xx.xx.xx*
Response	`dns server ip setting will change to: xx.xx.xx.xx`
Description	Configure the preferred DNS server's IP address for the controller's network port. The configuration will take effect immediately once the command is sent to the controller.

Example:

If you want to configure the preferred DNS server's IP address for the controller's network port as 202.96.134.133:

Command:

```
config set dnsserver ip4addr 202.96.134.133
```

Response:

```
dns server ip setting will change to: 202.96.134.133
```

2.1.39 config set dnsserver ip4addr2

Command	`config set dnsserver ip4addr2` *xx.xx.xx.xx*
Response	`dns server ip2 setting will change to: xx.xx.xx.xx`
Description	Configure the alternate DNS server's IP address for the controller's network port. The configuration will take effect immediately once the command is sent to the controller.

Example:

If you want to configure the alternate DNS server's IP address for the controller's network port as 202.96.128.68:

Command:

```
config set dnsserver ip4addr2 202.96.128.68
```

Response:

```
dns server ip2 setting will change to: 202.96.128.68
```

2.1.40 config get version

Command	`config get version`
Response	`API version:` *v#.#* `System version:` *v#.#.#(v#.#.#)*
Description	Obtains IP controller version information. **Note:** • This command is used to obtain IP controller version information, which can be used for troubleshooting.

	• IP controller version information contains API version, web console version and service version.

Example:

If you want to obtain IP controller version information:

Command:

```
config get version
```

Response:

```
API version: v1.2
System version: v3.0.2 (v1.5.4)
```

Note:

v1.2 is API version. v3.0.2 is web console version. v1.5.4 is service version.

2.1.41 config get devicelist

Command	`config get devicelist`
Response	`devicelist is` *`hostname1 hostname2...`*
Description	Obtains online device list. **Note:** • **hostname1** and **hostname2** are device names. • This command is used to get all online device names. • If you want to obtain a list consisting of device types and offline devices, you can use **config get devicejsonstring**.

Example:

If you want to obtain online device list:

Command:

```
config get devicelist
```

Response:

```
devicelist isDIPD-5100-341B228000BCDIPD-5100-341B22800490
```

Note:

The current online devices areDIPD-5100-341B228000BC andDIPD-5100-341B22800490.

2.1.42 config get ipsetting

Command	`config get ipsetting`
Response	`ipsetting is:ip4addr` *`xx.xx.xx.xx`* `netmask` *`xx.xx.xx.xx`* `gateway` *`xx.xx.xx.xx`*
Description	Obtains network settings in LAN(AV) port.

Example:

If you want to obtain network settings in LAN(AV) port:

Command:

```
config get ipsetting
```

Response:

```
ipsetting is:ip4addr 169.254.1.100 netmask 255.255.0.0 gateway 169.254.1.1
```

Note:

LAN(AV) port's IP address is 169.254.1.100, subnet mask is 255.255.0.0 and gateway is 169.254.1.1.

2.1.43 config get ipsetting2

Command	`config get ipsetting2`
Response	`ipsetting2 is:ip4addr` *`xx.xx.xx.xx`* `netmask` *`xx.xx.xx.xx`* `gateway` *`xx.xx.xx.xx`*
Description	Obtains network settings in LAN(C) port.

Example:

If you want to obtain network settings in LAN(C) port:

Command:

```
config get ipsetting2
```

Response:

```
ipsetting2 is:ip4addr 192.168.11.223 netmask 255.255.0.0 gateway 192.168.11.1
```

Note:

LAN(C) port's IP address is 192.168.11.223, subnet mask is 255.255.0.0 and gateway is 192.168.11.1.

2.1.44 config get name

Command	`config get name{alias|hostname}`
Response	`hostname'alias is xxxx`
Description	Obtains device name or its alias. Note: • You can use a device name to obtain its alias or vice versa. • **alias** is device alias. **hostname** is device name. • If you use a device name to obtain its alias which is not set, response is "NULL". • If **config get name** is used without parameters, response is all device names and their aliases.

Example 1:

If you want to obtainDIPE5100-341B22430115's alias:

Command:

```
config get name DIPE5100-341B22430115
```

Response:

```
DIPE5100-341B22430115's alias is testDIPE
```

Example 2:

If you want to obtainDIPE5100-341B22430225's alias which is not set:

Command:

```
config get name DIPE5100-341B22430225
```

Response:

```
DIPE5100-341B22430225's alias is NULL
```

Example 3:

If you want to obtain all device names and their aliases:

Command:

```
config get name
```

Response:

```
DIPE5100-341B22430115's alias is testDIPE
DIPE5100-341B22430225's alias is NULL
```

2.1.45 config get device info

Command	`config get device info hostname1 hostname2...`

Response	devices json info:
	{
	"devices":
	[
	{
	"aliasname" : "RX-1"
	"key11:"value11"
	"key12:"value12"
	...
	},
	{
	"key21:"value21"
	"key22:"value22"
	...
	}
	...
	]
	}

Description	Obtains device working parameters in real time. **Note:** • **hostname1** and **hostname2** are device names. • You can get one or multiple devices' working parametersat one time. • Alias name feature is added from the API v1.7 version • It may take some time for IP controller to get device information. The developer must consider this factor when programming the caller's code. • Working parameters use**Key:Value** format. **Key** is a parameter name and **value**

	is its value. For more information, see 3.1Device Info section.

Example 1:

If you want to obtain DIPE5100-341B22F32001's working parameters:

Command:

```
config get device info DIPE5100-341B22F32001
```

Response:

```
devices json info:
{
    "devices":
    [
        {
 "aliasname": "RX-1",
            "name":"DIPE5100-341B22F32001",
            "version":"v2.5.8",
            "ip_mode":"dhcp",
            "ip4addr":"169.254.107.239",
            "netmask":"255.255.0.0",
            "mac":"34:1b:22:f3:20:01",
            "gateway":" ",
            "hdcp":false,
            "sourcein":"hdmi",
            "enc_rc_mode":"vbr",
            "profile":"hp",
            "cbr_avg_bitrate":10000,
            "vbr_max_bitrate":20000,
            "vbr_min_qp":0,
            "vbr_max_qp":25,
            "fixqp_iqp":25,
            "fixqp_pqp":25,
            "enc_gop":60,
            "enc_fps":60,
            "transport_type":"raw"
        }
    ]
}
```

Example2:

If you want to obtainDIPD-5100-341B228007BD's working parameters:

Command:

```
config get device infoDIPD-5100-341B228007BD
```

Response:

```
devices json info:
{
    "devices":
    [
        {
            "name":"IPD5100-341B228007BD",
            "version":"v2.5.6",
            "ip_mode":"autoip",
            "ip4addr":"169.254.5.173",
            "netmask":"255.255.0.0",
            "mac":"34:1b:22:80:07:bd",
            "gateway":" ",
            "hdcp":false,
            "sourcein":"null",
            "audio":
            [
                {
                    "name":"lineout1",
                    "mute":false
                }
            ]
        }
    ]
}
```

Example 3:

If you want to obtain the working parameters of DIPE6000-D88039A4C559:

Command:

```
config get device info DIPE6000-D88039A4C559
```

Response:

```
 {
   "aliasname" : "tx6",
    "analog_audio_direction" : "INPUT",
   "bit_perpixel" : 8,
   "color_space" : "RGB",
```

```
   "edid"   :   "00ffffffffffff0010a6000156524c420816010380341d780a01c1a057479827124c
4c2108008140010101010101010101010101010101023a801871382d40582c460040846300001e08e80
030f2705a80b0588a00ba892100001e000000fc0044656661756c740a2020202020000000fd001855
0e853c000a202020202020016b020340735801020304051112131 41f2021220607905d5e5f6263646
061230d0707830f00006d030c001000383c20406801020367d85dc401788003e40f0000c0011d00bc
52d01e20b828554040846300001e023a80d072382d40102c458040846300001e023a801871382d405
82c450040846300001e000000000000000000f2",
   "gateway" : "10.0.254.253",
   "hdcp14_enable" : true,
   "hdcp22_enable" : true,
   "ip4addr" : "10.0.254.5",
   "ip_mode" : "dhcp",
   "name" : "DIPE6000-D88039A4C559",
   "netmask" : "255.255.0.0",
   "serial_param" : "57600-8n1",
   "stream0_enable" : false,
   "stream0fps_by2_enable" : false,
   "stream1_enable" : true,
   "stream1_scale" : "1280x1024",
   "stream1fps_by2_enable" : false,
   "temperature" : 59,
   "version" : "3.5.0.0",
   "video_input" : false,
   "video_source" : "hdmi",
   "video_timing" : "0x0@0"
  }
```


Note: The information of “color_space” and “bit_perpixel” have been added to the command response for IPX6000.

2.1.46 config get device status

Command	configget devicestatus*hostname1 hostname2...*
Response	devices status json info: { "devices_status": [{ "aliasname" : "TX-1", "name":"DIPE5100-341B22800BCC", " hdmi in active":"false", " resolution":" 0x0", " hdmi in frame rate":"0", " encoding enable":"true", " video stream ip address":" 0.0.0.0", " audio stream ip address":" 0.0.0.0",

	" line out audio enable":" true", " stream resolution":" 0x0", " stream frame rate":" 0", }] }
Description	Obtains device status in real time. **Note:** • **hostname1** and **hostname2** are device names. • Device status information uses json format. • Devices' status information depends on the device instead of IP controller, IP controller is only used for passing by.

Example:

If you want to obtain DIPE5100-341B22800BCC's status:

Command:

```
config get device status DIPE5100-341B22800BCC
```

Response:

```
devices status json info:
{
    "devices_status":
    [
        {
            "aliasname" : "TX-1",
            "name":"DIPE5100-341B22800BCC",
            " hdmi in active":"false",
            " resolution":" 0x0",
            " hdmi in frame rate":"0",
            " encoding enable":"true",
            " video stream ip address":" 0.0.0.0",
            " audio stream ip address":" 0.0.0.0",
            " line out audio enable":" true",
            " stream resolution":" 0x0",
            " stream frame rate":" 0",
        }
    ]
}
```

2.1.47 config get devicejsonstring

Command	config get devicejsonstring

Response	device json string: [
	{
	"aliasName" : "xxx",
	"deviceType" : "Transmitter/Receiver",
	"group" : [
	{
	"name" : "xxx",
	"sequence" : xxx
	}
	],
	"ip" : "xx.xx.xx.xx",
	"online" : true/false,
	"sequence" : xxx,
	"trueName" : "xxx"
	"previewurl" : "xxx.xxx.xxx.xxx"
	nameoverlay" : true/false
	}
	…
	]

Description	Obtains all device information. **Note:** • "aliasName" represents device alias name (If no alias name appears, it means that this device is not given an alias name). • "deviceType" represents device type: TX represents transmitter, RX represents receiver, TRX represents transceiver. • "group" represents a group. One RX unit can only be put in one group. "sequence" in "group" represents the position of this group, which starts with 1. If "sequence" is 0, it means that this group is not arranged in specific order. In this case, you can put this group in a position based on programming. • "ip" represents device IP address such as 169.254.5.24. • "online" represents device status, online or offline. "true" represents device is online. "false" represents device is offline. • "sequence" in a device represents the position of this device in its group, which starts with 1. If "sequence" is 0, it means that this device is not arranged in specific order. In this case, you can put this device in a position based on programming. • "trueName" represents device true name.

Example:

If you want to obtain all device information:

Command:

```
config get devicejsonstring
```

Response:

```
device json string: [
    {
        "aliasName" : "rx",
        "deviceType" : "Transmitter",
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 0
            }
        ],
        "ip" : "169.254.3.73",
        "online" : true,
        "sequence" : 1,
        "trueName" : "DIPE5100-341B22F40201"
        "previewurl" : "http://169.254.123.9/logo.jpg",
        "nameoverlay" : true
    }
…
]
```

2.1.48 config get scenejsonstring

Command	config get scenejsonstring

Response	scene json string:[
	{
	"group" : [
	{
	"name" : "xxx",
	"sequence" : xxx
	}
	],
	"layoutseq" : xxx,
	"m" : xxx,
	"n" : xxx,
	"name" : "xxx-xxx",
	"rxArray" : [
	[
	{

```
                    "aliasName" : "xxx",
                    "deviceType" : "Transmitter/Receiver",
                    "group" : [
                        {
                            "name" : "xxx",
                            "sequence" : xxx
                        }
                    ],
                    "online" : true/false,
                    "rxstatus" : xxx,
                    "sequence" : xxx,
                    "trueName" : "xxx",
                    "txName" : "xxx"
                },
                {
                    "aliasName" : "xxx",
                    "deviceType" : "Transmitter/Receiver",
                    "group" : [
                        {
                            "name" : "xxx",
                            "sequence" : xxx
                        }
                    ],
                    "online" : true/false,
                    "rxstatus" : xxx,
                    "sequence" : xxx,
                    "trueName" : "xxx",
                    "txName" : "xxx"
                }
            ],
            [
                {
                    "aliasName" : "xxx",
                    "deviceType" : "Transmitter/Receiver",
                    "group" : [
                        {
                            "name" : "xxx",
                            "sequence" : xxx
                        }
                    ],
                    "online" : true/false,
                    "rxstatus" : xxx,
                    "sequence" : xxx,
                    "trueName" : "xxx",
                    "txName" : "xxx"
                },
                {
                    "aliasName" : "xxx",
```

```
                    "deviceType" : "Transmitter/Receiver",
                    "group" : [
                        {
                            "name" : "xxx",
                            "sequence" : xxx
                        }
                    ],
                    "online" : true/false,
                    "rxstatus" : xxx,
                    "sequence" : xxx,
                    "trueName" : "xxx",
                    "txName" : "xxx"
                }
            ]
        ],
        "sceneAutoApply" : true/false,
        "sequence" : xxx,
        "txListArray" : [
            [
                {
                    "devices" : []
                },
                {
                    "devices" : []
                }
            ],
            [
                {
                    "devices" : []
                },
                {
                    "devices" : []
                }
            ]
        ],
        "vwConfigList" : [
            {
                "col_count" : xxx,
                "mode" : "xxx",
                "name" : "xxx",
                "oh" : xxx,
                "ow" : xxx,
                "pos_col" : xxx,
                "pos_row" : xxx,
                "row_count" : xxx,
                "vh" : xxx,
                "vw" : xxx
            },
```

	<pre>{
	"col_count" : xxx,
	"mode" : "xxx",
	"name" : "xxx_xxx",
	"oh" : xxx,
	"ow" : xxx,
	"pos_col" : xxx,
	"pos_row" : xxx,
	"row_count" : xxx,
	"vh" : xxx,
	"vw" : xxx
	}
	]
	}
	]</pre>

Description	Obtains all scene information. **Note:** • "group" represents a group. One scene can only be put in one group. "sequence" in" group" represents the position of this group, which starts with 1. If "sequence" is 0, it means that this group is not arranged in specific order. In this case, you can put this group in a position based on programming. • "layoutseq" represents the position of this scene in video wall. • "n" and "m" represent the number of rows and columns respectively in a scene. • "name" represents scene name, such as s • "rxArray" describes RX in a form of two-dimensional array in a scene. • "sequence" in a scene represents the position of video wall which contains this scene, which starts with 1. If "sequence" is 0, it means that this video wall is not arranged in specific order. In this case, you can put it in a position based on programming. • "txListArray" describes TX in a form of two-dimensional array in a scene. • "vwConfigList" represents the configuration of combination screen in a scene. "name" represents combination screen name, which uses "scene name_ combination screen name" in IP controller (DSC010)."pos_row" represents the start place of the first row."pos_col" represents the start place of the first column."row_count" represents the number of rows in combination screen."col_count"represents the number of columns in combination screen.

2.1.49 config get telnet alias

Command	`config get telnet alias`
Response	`telnet alias is {on|off}`
Description	Get Telnet session alias mode.

Example:

If you want to get the telnet alias mode:

Command:

```
config get telnet alias
```

Response:

```
telnet alias is off
```

2.1.50 config get rs-232 alias

Command	`config get rs-232 alias`
Response	`rs-232 alias is {on|off}`
Description	Get the rs-232 alias mode.

Example:

If you want to get the uart alias mode:

Command:

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

Response:

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

2.1.51 config getsystem sshservice

Command	`config get system sshservice`
Response	`system sshservice is {on|off}`
Description	Get the system SSH service mode.

Example:

If you want to get the system SSH service mode:

Command:

```
config get system sshservice
```

Response:

```
system sshservice is on
```

2.1.52 configure get system info

Command	`config get system info`

Response	system info:
	{
	"meminfo" : {
	"total" : 244292,
	"used" : 232848,
	"free" : 11444,
	"shared" : 0,
	"buffers" : 68616,
	"cached" : 83440

	}, "cpuinfo" : { "user" : 3.4, "sys" : 9.3, "idle" : 87.0, "wait" : 0.1, "hi" : 0.0, "si" : 0.1 } }
Description	Get the system status information, including CPU and memory usage.

Example:

If you want to get the system status information:

Command:

```
config get system info
```

Response:

```
system info:
{
    "meminfo" : {
        "total" : 244292,
        "used" : 232848,
        "free" : 11444,
        "shared" : 0,
        "buffers" : 68616,
        "cached" : 83440
    },
    "cpuinfo" : {
        "user" : 3.4,
        "sys" : 9.3,
        "idle" : 87.0,
        "wait" : 0.1,
        "hi" : 0.0,
        "si" : 0.1
    }
}
```

2.1.53 config get system ldap

Command	config get system ldap

Response	Returns LDAP configuration information of the IP controller.
Description	Obtains LDAP configuration information of the IP controller.

Example:

If you want to obtain LDAP configuration information of the IP controller:

Command:

```
config get system ldap:
```

Response:

```
LDAP config json info:
{
    "802_1X_status":
    [
        {
            "ldap_status": "on",
           "ldap_uri": "ldaps://192.168.17.225:636",
"ldap_uid": " jeremy",
           "ldap_base_dn": "dc=my-domain, dc=com",
           "ldap_bind_dn": "cn=jeremy, ou=People, dc=my-domain, dc=com",
           "ldap_attr": "uid",
        }
    ]
}
```

2.1.54 config get system 802_1x

Command	`config get system 802_1x`
Response	Returns 802.1x configuration information of the IP controller.
Description	Obtains 802.1x configuration information of the IP controller.

Example:

If you want to configure 802.1x for the IP controller:

Command:

```
config get system 802_1x
```

Response:

```
802_1x info:
[
    {
        "ieee802_1x_enable": "true",
       "ieee802_1x_mode": "mschapv2/tls"
```

```
        "ieee802_1x_mschapv2_user": "user",
"ieee802_1x_mschapv2_password": " password",
        "ieee802_1x_tls_user": "user",
       "ieee802_1x_tls_private_key_password”:”password”;
       “ieee802_1x_ca_mode”:”true” // Authentication status of CA certificate
    }
]
```

2.1.55 config get system lcd

Command	`config get system lcd`
Response	`config get system lcd {ipversion|logo|off}`
Description	Obtain the LCD display's configuration information of the IP controller. • ipversion: Displays IP address and firmware version. • logo: Displays logo. • off: LCD display is turned off. By default, the configuration is ipversion.

Example:

If you want to obtain the configuration information of the controller's LCD display:

Command:

```
config get system lcd
```

Response:

```
config get system lcd ipversion
```

2.1.56 config get controller info

Command	`config get controller info`

Response	`controller info:`
	`[`
	`{`
	`"hostname_av" : "DSC010-6EFAB4238C66",`
	`"hostname_ctl" : "DSC010-2A14823D757B",`
	`"mac_av" : "6e:fa:b4:23:8c:66",`
	`"mac_ctl" : "2a:14:82:3d:75:7b",`
	`"serialNumber" : "12345678912345",`
	`"version" : "v1.0.5"`
	`}`
	`]`

Description	Obtain the controller's information, including hostname, mac address, serial no., version information, etc.

Example:

Command:

```
config get controller info
```

Response:

```
controller info:
[
    {
        "hostname_av" : "DSC010-6EFAB4238C66",
        "hostname_ctl" : "DSC010-2A14823D757B",
        "mac_av" : "6e:fa:b4:23:8c:66",
        "mac_ctl" : "2a:14:82:3d:75:7b",
        "serialNumber" : "12345678912345",
        "version" : "v1.0.5"
    }
]
```

2.1.57 config get dnsserver ip4addr

Command	config get dnsserver ip4addr
Response	dns server ip4addr is:xx.xx.xx.xx
Description	Obtain the preferred DNS server's IP address for the controller's network port.

Example:

Command:

```
config get dnsserver ip4addr
```

Response:

```
dns server ip4addr is:202.96.134.133
```

2.1.58 config get dnsserver ip4addr2

Command	config get dnsserver ip4addr2
Response	dns server ip4addr2 is:xx.xx.xx.xx
Description	Obtain the alternate DNS server's IP address for the controller's network port.

Example:

Command:

```
config get dnsserver ip4addr2
```

Response:

```
dns server ip4addr2 is:202.96.128.68
```

2.1.59 config set system realtime

Command	config set system realtime DD-MM-YY HH:MM:SS
Response	config set system realtime DD-MM-YY HH:MM:SS
Description	Configure device time

Example:

Command:

```
config set system realtime 27-01-2024 18:41:50
```

Response:

```
config set system realtime 27-01-2024 18:41:50
```

2.1.60 config get system realtime

Command	config get system realtime
Response	RealTime:DD-MM-YY HH:MM:SS
Description	Get device time

Example:

Command:

```
config get system realtime
```

Response:

```
RealTime:Sat,27-01-2024,18:45:10
```

2.1.61 config get system runtimesincereboot

Command	config get system runtimesincereboot
Response	Runtime since reboot: xx days HH:MM
Description	Get runtime after last reboot

Example:

Command:

```
config get system runtimesincereboot
```

Response:

```
Runtime since reboot:4 days 18:45
```

2.2 matrix Commands

2.2.1 matrix set

Command	matrix set *TX1 RX1 RX2, TX2 RX3 RX4, ...*
Response	matrix set *TX1 RX1 RX2, TX2 RX3 RX4, ...*
Description	• Controls the switching of RX to TX. • Parameters are separated by commas such as segments TX1 RX1 RX2, TX2 RX3 RX4. Every segment starts with TX and is followed by some RX which are switched to this TX. If a segment starts with TX whose name is "NULL" the followed RX will not decode video. "NULL" is not case sensitive. • For RX in video wall, this command is used to switch to another TX but will not clear video wall settings. If a RX in video wall displays a certain position of TX1's video, after this RX is switched to TX2, RX will still display the same position of TX2's video. Other RX in video wall functions in the same way. • For RX supporting multi-view, this command is used to switch to another TX for full-screen displaying.

Example1:

If you want RXDIPD-5100-341B22800316 andDIPD-5100-341B22800309 to be switched to TX DIPE5100-341B22FFFFC1, RXDIPD-5100-341B22800319 to TX DIPE5100-341B22FFFFC2, and RXDIPD-5100-341B2280031A to TX DIPE5100-341B22FFFFC3:

Command:

```
matrix set DIPE5100-341B22FFFFC1 IPD5100-341B22800316 IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-341B22FFFFC3DIPD-5100-341B2280031A
```

Response:

```
matrix set DIPE5100-341B22FFFFC1 IPD5100-341B22800316 IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-341B22FFFFC3DIPD-5100-341B2280031A
```

Example2:

If you want RXDIPD-5100-341B22800316 to stop decoding video:

Command:

```
matrix set NULLDIPD-5100-341B22800316
```

Response:

```
matrix set NULLDIPD-5100-341B22800316
```

2.2.2 matrix get

Command	matrix get
Response	matrix information: *TX1 RX1* *TX2 RX3* *TX2 RX4*

	...
Description	Obtains TX played by RX in matrix. **Note:** • For video wall, the response contains RX and its linked TX but does include video wall information. If you want to obtain video wall information, you can use **vw** command. • If TX is NULL, RX does not decode video. "NULL" is not case sensitive. • Response does not include RX which supports multi-view.

Example:

If you want to obtain TX played by RX in matrix:

Command:

```
matrix get
```

Response:

```
matrix information:
DIPE5100-341B2243011ADIPD-5100-341B22800BCD
DIPE5100-341B2243011ADIPD-5100-341B22800BCE
DIPE5100-341B2243011ADIPD-5100-341B22800BCA
nullDIPD-5100-341B22800BC6
```

Note:

The response indicates thatDIPD-5100-341B22800BCD,DIPD-5100-341B22800BCE, andDIPD-5100-341B22800BCA all play DIPE5100-341B2243011A, andDIPD-5100-341B22800BC6 does not decode video.

2.2.3 matrix video set

Command	`matrix video set TX1 RX1 RX2,TX2 RX3 RX4,...`
Response	`matrix video set TX1 RX1 RX2,TX2 RX3 RX4,...`
Description	Change the RX and TX video matrix link relationship. Note: Video matrix switch, that a certain or a few RX are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. Such as NULL for TX at the front of RX, mean RX get not TX link on

Example:

If you want to change or make the TX video link to any of RX in matrix:

Command:

```
matrix video set DIPE5100-341B22FFFFC2DIPD-5100-341B22800316, DIPE5100-
341B22FFFFC2
IPD5100-341B22800309,    DIPE5100-341B22FFFFC2DIPD-5100-341B22800319,    DIPE5100-
341B22FFFFC2DIPD-5100-341B2280031A
```

Response:

```
matrix    video    set    DIPE5100-341B22FFFFC2DIPD-5100-341B22800316,    DIPE5100-
341B22FFFFC2DIPD-5100-341B22800309,   DIPE5100-341B22FFFFC2DIPD-5100-341B22800319,
DIPE5100-341B22FFFFC2DIPD-5100-341B2280031A
```

2.2.4 matrix video get

Command	matrix video get RX1 RX2……
Response	matrix video information: TX1 RX1 TX2 RX3 TX2 RX4 ...
Description	Obtain all or parts of RX video matrix information Note: If the command don't get the RX, it means obtain all RX video matrix information.

Example:

If you want to obtain all RX video matrix information:

Command:

```
matrix video get
```

Response:

```
matrix video information:

DIPE5100-341B2243011ADIPD-5100-341B22800BCD

DIPE5100-341B2243011ADIPD-5100-341B22800BCE

DIPE5100-341B2243011ADIPD-5100-341B22800BCA

DIPE5100-341B2243011ADIPD-5100-341B22800BC6
```

2.2.5 matrix audio set

Command	matrix audio set TX1 RX1 RX2,TX2 RX3 RX4,...
Response	matrix audio set TX1 RX1 RX2,TX2 RX3 RX4,...
Description	Change or set the audio matrix link relationship between TX and RX. Note: Audio matrix switch, that a certain or a few RX are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. Such as NULL for TX at the front of RX, mean RX get not TX link on

Example:

If you want to change or make the TX audio link to any of RX in matrix:

Command:

```
matrixaudio set DIPE5100-341B22FFFFC2DIPD-5100-341B22800316, DIPE5100-
341B22FFFFC2

IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-

341B22FFFFC2DIPD-5100-341B2280031A
```

Response:

```
matrixaudio set DIPE5100-341B22FFFFC2DIPD-5100-341B22800316, DIPE5100-
341B22FFFFC2

IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-

341B22FFFFC2DIPD-5100-341B2280031A
```

2.2.6 matrix audio get

Command	`matrixaudio get RX1 RX2……`

Response	`matrix audio information:`
	`TX1 RX1`
	`TX2 RX3`
	`TX2 RX4`
	`...`

Description	Obtain all or parts of RX audio matrix information Note: If the command doesn't get the RX, it means obtain all RX audio matrix information.

Example:

If you want to obtain all RX audio matrix information:

Command:

```
matrix audio get
```

Response:

```
matrix audio information:
DIPE5100-341B2243011ADIPD-5100-341B22800BCD
DIPE5100-341B2243011ADIPD-5100-341B22800BCE
DIPE5100-341B2243011ADIPD-5100-341B22800BCA
DIPE5100-341B2243011ADIPD-5100-341B22800BC6
```

2.2.7 matrix usb set (for DIPX-5100/DIPX-6000)

Command	`matrix usb set TX1 RX1 RX2, TX2 RX3 RX4, ...`
Response	`matrix usb set TX1 RX1 RX2, TX2 RX3 RX4, ...`
Description	Change or set the usb matrix link relationship between TX and RX Note: • USB matrix switch, that a certain or a few RX are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. Such as NULL for TX at the front of RX, mean RX get not TX link on • For IPX6000 devices, this command just controls the ICRON USB extension feature, furthermore, a TX can be paired with only one RX.

Example:

If you want to change or make the TX usb link to any of RX in matrix:

Command:

```
matrix usb set DIPE5100-341B22FFFFC2DIPD-5100-341B22800316, DIPE5100-341B22FFFFC2
IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-
341B22FFFFC2DIPD-5100-341B2280031A
```

Response:

```
matrix usb set DIPE5100-341B22FFFFC2DIPD-5100-341B22800316, DIPE5100-341B22FFFFC2
IPD5100-341B22800309, DIPE5100-341B22FFFFC2DIPD-5100-341B22800319, DIPE5100-
341B22FFFFC2DIPD-5100-341B2280031A
```

2.2.8 matrix usb get (for DIPX-5100/DIPX-6000)

Command	`matrix usb get RX1 RX2 ……`

Response	`matrix usb information:`
	`TX1 RX1`
	`TX2 RX3`
	`TX2 RX4`
	`...`

Description	Obtain all or parts of RX usb matrix information Note: If the command doesn't include any RX, all RX usb matrix information will return.

Example:

If you want to obtain all RX usb matrix information:

Command:

```
matrix usb get
```

Response:

```
matrix usb information:
DIPE5100-341B2243011ADIPD-5100-341B22800BCD
DIPE5100-341B2243011ADIPD-5100-341B22800BCE
DIPE5100-341B2243011ADIPD-5100-341B22800BCA
DIPE5100-341B2243011ADIPD-5100-341B22800BC6
```

2.2.9 matrix usb_hid set (For DIPX-6000)

Command	`matrix usb_hid set` *`txdev mode [rxdev]`*
Response	`matrix usb_hid set` *`txdev mode [rxdev]`*
Description	For USB_HID matrix routing: route the USB_HID input of txdev to the target device. Four modes are provided: {single|api|all|null} • single: rxdev is available in this mode. Route the USB_HID input of txdev to rxdev. • api: route the USB_HID input of txdev to API server (IP controller). • all: route the USB_HID input of txdev to all IP6000 devices. • null: stop routing the USB_HID input of txdev.

Example:

Command:

```
matrix usb_hid set TX1 single RX2
```

Response:

```
matrix usb_hid set TX1 single RX2
```

2.2.10 matrix usb_hid get (for DIPX-6000)

Command	`matrix usb_hid get` *dev…*

Response	`matrix usb_hid information:`
	`dev mode [rxdev]`
	`...`

Description	Get the USB_HID routing information for all/part of the devices. If the command doesn't include any RX, all RXs' routing information will return; if yes, the RX's routing information will return.

Example:

Command:

```
matrix usb_hid get
```

Response:

```
matrix usb_hid information:
tx1 single tx2
tx2 api
tx3 all
tx4 null
```

2.2.11 matrix audio2 set (for DIPX-6000)

Command	`matrix audio2 set` *TX1 RX1 RX2,TX2 RX3 RX4,...*
Response	`matrix audio2 set` *TX1 RX1 RX2,TX2 RX3 RX4,...*
Description	Change or set the analog audio matrix link relationship between TX and RX. **Note**: Analog audio matrix switch; a certain or a few RXs are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. If TX is NULL, it means the link relationship between RX and its corresponding TX is cleared.

Example:

If you want to set analog audio link relationship between RX and

TX, **Command:**

```
matrix audio2 set TX1 RX1 RX2
```

Response:

```
matrix audio2 set TX1 RX1 RX2
```

Command:

```
matrix audio2 set TX1 RX1 RX2, TX2 RX3 RX4
```

Response:

```
matrix audio2 set TX1 RX1 RX2, TX2 RX3 RX4
```

2.2.12 matrix audio2 get (for DIPX-6000)

Command	`matrix audio2 get` *`RX1 RX2`*

Response	`matrix audio2 information:`
	`TX1 RX1`
	`TX2 RX3`
	`TX2 RX4`
	`...`

Description	Obtain all or parts of analog audio matrix information for RX. **Note**: If the command doesn't include the RX, it means obtain all RXs' analog audio matrix information.

Example:

If you want to obtain all RXs' analog audio matrix information:

Command:

```
matrix audio2 get
```

Response:

```
matrix audio2 information:
TX1 RX1
TX1 RX2
```

2.2.13 matrix audio3 set (for DIPX-5100/DIPX-5100L)

Command	`matrix audio3 set` *`TX1 RX1 RX2, TX2 RX3 RX4, ...`*
Response	`matrix audio3 set` *`TX1 RX1 RX2, TX2 RX3 RX4, ...`*
Description	Set the link relationship between RX and TX in the ARC audio matrix. **Note**: Set the link relationship for the ARC audio matrix; a certain or multiple TXs are linked to a certain RX. Each record of RX and its associated TX(s) will be separated by a comma. If RX is NULL, it means the link relationship between TX and its corresponding RX is cleared.

Example 1:

If you want to route ARC audio from RX1 to TX1 and TX2:

Command:

```
matrix audio3 set RX1 TX1 TX2
```

Response:

```
matrix audio3 set RX1 TX1 TX2
```

Example 2:

If you want to route ARC audio from RX1 to TX1 and TX2, and from RX2 to TX3 and TX4:

Command:

```
matrix audio3 set RX1 TX1 TX2, RX2 TX3 TX4
```

Response:

```
matrix audio3 set RX1 TX1 TX2, RX2 TX3 TX4
```

2.2.14 matrix audio3 get (for DIPX-5100/DIPX-5100L)

Command	`matrix audio3 get` *TX1 TX2*

Response	`matrix audio3 information:`
	`RX1 TX1`
	`RX2 TX2`
	`RX2 TX4`
	`...`

Description	Get which RX is linked to the specified TX(s) in the ARC audio matrix. **Note**: If one or more TXs are specified, the controller will return the link relationship information of the TXs and the corresponding RX(s); if not, the controller will return the link relationship information of all TXs and the corresponding RX(s).

Example 1:

If you want to get the link relationship of all TXs and the associated RXs in the ARC audio matrix:

Command:

```
matrix audio3 get
```

Response:

```
matrix audio3 information
RX1 TX1
RX1 TX2
```

2.2.15 matrix infrared set (for DIPX-5100)

Command	`matrix infrared set TX1 RX1 RX2,TX2 RX3 RX4,...`
Response	`matrix infrared set TX1 RX1 RX2,TX2 RX3 RX4,...`
Description	Change or set the infrared matrix link relationship between TX and RX. Note: Infrared matrix switch, a certain or a few RX are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. Such as NULL for TX at the front of RX, mean RX get not TX link on

Example:

If you want to change or make the TX infrared link to any of RX in matrix:

Command:

```
matrix infrared set TX1 RX1 RX2
```

Response:

```
matrix infrared set TX1 RX1 RX2
```

2.2.16 matrix infrared get (for DIPX-5100)

Command	`matrix infrared get RX1 RX2`

Response	`matrix infrared information:`
	`TX1 RX1`
	`TX2 RX3`
	`TX2 RX4`
	`...`

Description	Obtain all or parts of RX infrared matrix information. Note: If the command doesn't include any RX, it means obtain all RX infrared matrix information.

Example:

If you want to obtain all RX infrared matrix information:

Command:

```
matrix infrared get
```

Response:

```
matrix infrared information:
DIPE5100-341B2243011ADIPD-5100-341B22800BCD
DIPE5100-341B2243011ADIPD-5100-341B22800BCE
DIPE5100-341B2243011ADIPD-5100-341B22800BCA
DIPE5100-341B2243011ADIPD-5100-341B22800BC6
```

2.2.17 matrix infrared2 set (only for DIPX-6000)

Command	`matrix infrared2 set` *`txdev mode [rxdev]`*
Response	`matrix infrared2 set`*`txdev mode [rxdev]`*
Description	Change or set the infrared matrix link relationship between TX and the target device. There are four modes: single, api, all and null. • single: RX is available, set the infrared link between TX and RX. • api: set the infrared link between TX and API server (i.e. DSC010). • all: set the infrared link between TX and all IP6000 devices. • null: stop infrared output of TX.

Example:

If you want to change or set the TX infrared link to a device:

Command:

```
matrix infrared2 set TX1 single RX2
```

Response:

```
matrix infrared2 set TX1 single RX2
```

2.2.18 matrix infrared2 get (only for DIPX-6000)

Command	`matrix infrared2 get` *dev…*
Response	`matrix infrared2 information:` `dev mode [rxdev]`
Description	Obtain the infrared matrix information of all or parts of the devices. Note: If the command doesn't include the device's name, it means to obtain the infrared matrix information of all devices.

Example:

If you want to obtain all devices' infrared matrix information:

Command:

```
matrix infrared2 get
```

Response:

```
matrix infrared2 information:
tx1 single tx2
tx2 api
tx3 all
tx4 null
```

2.2.19 matrix serial set (for DIPX-5100)

Command	`matrix serial set TX1 RX1 RX2,TX2 RX3 RX4,...`
Response	`matrix serial set TX1 RX1 RX2,TX2 RX3 RX4,...`
Description	Change or set the serial matrix link relationship between TX and RX. Note: Serial matrix switch, a certain or a few RX are linked by a certain or some TX. Each record of TX and its associated RX will be separated by a comma. Such as NULL for TX at the front of RX, mean RX get not TX link on.

Example:

If you want to change or make the TX serial link to any of RX in matrix:

Command:

```
matrix serial set TX1 RX1 RX2
```

Response:

```
matrix serial set TX1 RX1 RX2
```

2.2.20 matrix serial get (for DIPX-5100)

Command	`matrix serial get RX1 RX2`

Response	`matrix serial information:`
	`TX1 RX1`
	`TX2 RX3`
	`TX2 RX4`
	`...`

Description	Obtain all or parts of RX serial matrix information. Note: If the command doesn't include any RX, it means obtain all RX serial matrix information.

Example:

If you want to obtain all RX serial matrix information:

Command:

```
matrix serial get
```

Response:

```
matrix serial information:
DIPE5100-341B2243011ADIPD-5100-341B22800BCD
DIPE5100-341B2243011ADIPD-5100-341B22800BCE
DIPE5100-341B2243011ADIPD-5100-341B22800BCA
DIPE5100-341B2243011ADIPD-5100-341B22800BC6
```

2.2.21 matrix serial2 set (only for DIPX-6000)

Command	`matrix serial2 set` *`txdev mode [rxdev]`*
Response	`matrix serial2 set` *`txdev mode [rxdev]`*
Description	Change or set the serial matrix link relationship between TX and the target device. There are four modes: single, api, all and null. • single: RX is available, set the serial link between TX and RX. • api: set the serial link between TX and API server (i.e. DSC010). • all: set the serial link between TX and all IP6000 devices. • null: stop serial link between TX and other devices.

Example:

If you want to change or make the TX serial link to any device in matrix:

Command:

```
matrix serial2 set TX1 single RX2
```

Response:

```
matrix serial2 set TX1 single RX2
```

2.2.22 matrix serial2 get (only for DIPX-6000)

Command	`matrix serial2 get dev...`

Response	`matrix serial2 information:`
	`dev mode [rxdev]`
	`...`

Description	Obtain the serial matrix information of all or parts of the devices. Note: If the command doesn't include the device's name, it means to obtain the serial matrix information of all the devices.

Example:

If you want to obtain all devices' serial matrix information:

Command:

```
matrix serial2 get
```

Response:

```
matrix serial2 information:
tx1 single tx2
tx2 api
tx3 all
tx4 null
```

2.3 vw Commands

2.3.1 vw add

Command	`vw add vw-name n m TX`
Response	`videowall item vw-name create and assign TX to it`
Description	Creates an n x m video wall configuration and assigns a TX. **Note:** • **vw-name** is video wall name and is different from others. • n is the number of the row, m is the number of the column. • This command is used to create records in IP controller but does not change devices' working status, for example devices still work as they were.

Example:

If you want to create a 2 x 2 video wall configuration **vwtest1** and assign TX DIPE5100-341B2243011A:

Command:

```
vw add vwtest1 2 2 DIPE-5100-341B2243011A
```

Response:

```
videowall item vwtest1 create and assign DIPE5100-341B2243011A to it
```

2.3.2 vw rm

Command	`vw rm vw-name`
Response	`videowall item vw-name removed`
Description	Removes a video wall configuration. **Note:** • **vw-name** is video wall name. • This command is used to remove records of video wall configuration in IP controller but does not change devices' working status. If the current video wall is removed using this command, RX in this video wall still plays its previous picture.

Example:

If you want to remove a video wall configuration **vwtes1**:

Command:

```
vw rm vwtest1
```

Response:

```
videowall item vwtest1 removed
```

2.3.3 vw rm vwname rx

Command	`vw rm vw-name rx1 rx2…`
Response	`videowall config change: remove rx1 rx2… from vw-name`
Description	Removes one or multiple RX from video wall. If RX is removed, it displays an entire picture of TX.

Example:

If you want to remove RXDIPD-5100-341B22800BCE andDIPD-5100-341B22800BCA from video wall **vwtest1**:

Command:

```
vw rm vwtest1 DIPD-5100-341B22800BCEDIPD-5100-341B22800BCA
```

Response:

```
videowall config change: remove DIPD-5100-341B22800BCEDIPD-5100-341B22800BCA from
vwtest1
```

2.3.4 vw add position

Command	`vw add vw-name RX1 x1 y1 RX2 x2 y2...`

Response	`videowall item vw-name configuration added:`
	`RX1 x1 y1`
	`RX2 x2 y2`
	`...`

Description	Adds RX to a video wall configuration. Once this command is executed, RX will play video

	wall. **Note:** • **vw-name** is the name of the video wall. • Parameters contain segments like RX1 x1 y1 RX2 x2 y2. Every segment starts with RX and is followed by its position like 1 2, adding this RX to the first row and second column of video wall. Segments x1, y1, x2, y2 start from 1.

Example:

If you want to add four RX to a video wall configuration **vwtest2**:

Command:

```
vw add vwtest2DIPD-5100-341B22800BCD 1 1DIPD-5100-341B22800BC6 1 2DIPD-5100-
341B22800BCE 2 1 DIPD-5100-341B22800BCA 2 2
```

Response:

```
videowall item vwtest2 configuration added:
DIPD5100-341B22800BCD 1 1
DIPD5100-341B22800BC6 1 2
DIPD5100-341B22800BCE 2 1
DIPD5100-341B22800BCA 2 2
```

2.3.5 vw add layout

Command	vw add *vw-name* layout *n m TX RX11 RX12 RX13 RX1m RX21 ... RXnm*
Response	videowall*vw-name* layout *n*m TXRX11 RX12 RX13 RX1m RX21... RXnm*
Description	Creates an n x m video wall configuration, assigns TX and n x m RX to it. Once this command is executed, RX will play video wall. **Note:** • **vw-name** is video wall name. • n is the number of row. m is the number of column. • Parameters RX11 RX12 RX13…RX1m RX21…RXnm are RX and are automatically assigned positions in the video wall in order. RX11 RX12 . . . RX1m RX21 RX22 . . . RX2m ⋮ ⋮ ⋮ ⋮ RXn1 RXn2 . . . RXnm

Example:

If you want to create a 2x2 video wall configuration **vwtest3** which contains one TX DIPE5100-341B22430115 and four RX IPD500-341B22800BCD, IPD500-341B22800BC6, IPD500-341B22800BCE and IPD500-341B22800BCA:

Command:

```
vw add vwtest3 layout 2 2 DIPE5100-341B22430115 DIPD5100-
341B22800BCDDIPD5100-341B22800BC6 DIPD-5100-341B22800BCE DIPD-5100-341B22800BCA
```

Response:

```
videowall vwtest3 layout 2*2 DIPE5100-341B22430115DIPD-5100-341B22800BCDDIPD-
5100-341B22800BC6DIPD-5100-341B22800BCEDIPD-5100-341B22800BCA
```

2.3.6 vw change rx tx

Command	`vw change` *`RX TX`*
Response	`videowall config clear:` *`rxhostname`* `and connect to` *`txhostname`*
Description	Removes one RX from video wall and switch this RX to another TX to play its entire picture. **Note:** If TX is "NULL", RX will not decode video. "NULL" is not case sensitive.

Example:

If you want to remove RXDIPD-5100-341B22800BCA from video wall and switch this RX to TX DIPE5100-341B22430115 to play its entire picture:

Command:

```
vw change DIPD-5100-341B22800BCA DIPE-5100-341B22430115
```

Response:

```
videowall config clear:DIPD-5100-341B22800BCA and connect to DIPE5100-
341B22430115
```

2.3.7 vw change vw-name tx

Command	`vw change` *`vw-name TX`*
Response	`videowall` *`vw-name`* `tx connect to` *`txhostname`*
Description	Switchesto another sourcefor video wall.When this command is executed, video wall will play this TX. **Note:** • **vw-name** is video wall name. • If tx is "NULL", all RX will stop decoding video but video wall configuration does not change. "NULL" is not case sensitive.

Example:

If you want to switch to TX DIPE5100-341B22430115for video wall **vwtest2**:

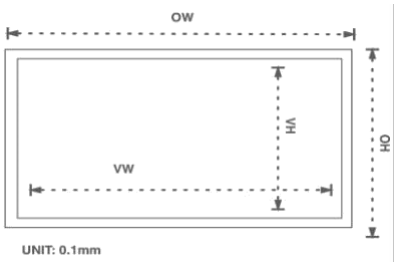
Command:

```
vw change vwtest2 DIPE5100-341B22430115
```

Response:

```
videowall vwtest2 tx connect to DIPE5100-341B22430115
```

2.3.8 vw bezelgap

Command	`vw bezelgap vw-name ow oh vw vh`
Response	`videowall vw-name's bezelgap: xx xx xx xx`
Description	Sets bezel compensation parameters. RX will trim video based on these parameters for bezel compensation, reducing the overall visual distortion of video wall. However, small parts of the video will be lost due to trimming. • **vw-name** is video wall name. • Units are in mm (0.1mm). • ow& oh= overall width & height of display device including the bezel. vw& vh = screen width &height.  • By defaults, RX takes ow and vw, oh and vh as being the same, and bezel as zero.

Example:

If you want to set bezel compensation parameters with outside size 194mm (ow) x110mm (oh) and screen size 192mm (vw)x108mm (vh) in video wall **vwtest5**:

Command:

```
vw bezelgap vwtest5 1940 1100 1920 1080
```

Response:

```
videowall vwtest5's bezelgap: 1940 1100 1920 1080
```

2.3.9 vw get

Command	`vw get`

Response	`Video wall information:`
	`vw-name1 tx1`
	`Row 1:` *`Rx1-11 Rx1-12`*
	`Row 2:` *`Rx1-21 Rx1-22`*
	`...`
	`vw-name2 tx2`
	`Row 1:` *`Rx2-11 Rx2-12`*
	`Row 2:` *`Rx2-21 Rx2-22`*
	`...`

Description	Obtains a list of all video walls.

	Note: • **vw-name1** and **vw-name2** are video wall names. • **tx1** is TX name of video wall **vw-name1**. **Rx1-11**, **Rx1-12**, **Rx1-21**and **Rx1-22** are RX of video wall **vw-name1**. Numbers like "11" in **Rx1-11** and "12" in **Rx1-12** are RX's positions in video wall. Other TX and RX are similar.

Example:

If you want to obtain a list of all video walls:

Command:

```
vw get
```

Response:

```
Video wall information:
Vm1 DIPE5100-341B22FFFFC2
Row 1:DIPD-5100-341B22FFFFD1DIPD-5100-341B22800309
Row 2:DIPD-5100-341B2280031ADIPD-5100-341B22800319
Vm2 DIPE5100-341B22FFFFC9
Row 1:DIPD-5100-341B2280031ADIPD-5100-341B22800319
Row 2:DIPD-5100-341B22FFFFD1DIPD-5100-341B22800309
```

2.3.10 vw stretch vw-name type

Command	`vw stretch vw-name type`
Response	`vw stretch vw-name type success`
Description	Set the stretch mode of the video wall. Three options of stretch mode are available as follows; and the default setting is stretch. • **fit:** The picture will scale in proportion; it will be displayed proportionally in maximized state; there may be blank space. • **stretch:** The picture will scale out of proportion; it will be stretched and shown according to the screen resolution; there's no blank space. • **fill:** The picture will scale in proportion to fill the screen; there's no blank space, while part of the picture may not be displayed. **Note:** • **vw-name** is the video wall name. • **Type** is the value of the stretch mode.

Example:

If you want to set the stretch mode for the video wall vwtest2,

Command:

```
vw stretch vwtest2 stretch
```

Response:

```
vw stretch vwtest2 stretch success
```

2.4 scene Commands

2.4.1 scene get

Command	`scene get`
Response	`scene list:` `scenename1 scenename2 scenename3…`
Description	Obtains all scene names.

Example:

If you want to obtain all scene names:

Command:

```
scene get
```

Response:

```
scene list:
Office-MeetingRoom Office-TrainingRoom Office-TeaRoom
```

2.4.2 scene active

Command	`scene active` *`scenename`*
Response	`scene` *`scenename`*`active success`
Description	Enables a new scene in video wall. This action takes effect immediately.

Example:

If you want to enable a new scene Office-Meeting Room in video wall:

Command:

```
scene active Office-MeetingRoom
```

Response:

```
scene Office-MeetingRoom active success
```

2.4.3 scene set

Command	`scene set` *`scenename posX posY tx1…`*
Response	`scene scenename's source in` *`[posX,posY] change to tx1`*
Description	Assigns a source to RX in a scene of video wall. This action makes RX display this source until **scene active *scenename*** is executed.

Example:

If you want to assign a source (tx1) to RX in scene Office-MeetingRoom of video wall:

Command:

```
scene set Office-MeetingRoom 1 2 tx1
```

Response:

```
scene Office-MeetingRoom's source in [1 2] change to tx1
```

2.4.4 scene change scenename txname

Command	scene change *scenename txname*
Response	scene scenename's tx change to tx1
Description	Assigns a source to all RX in a scene of video wall. This action makes all RX display this source until **scene active *scenename*** is executed.

Example:

If you want to assign a source (tx1) to all RX in scene1 of video wall:

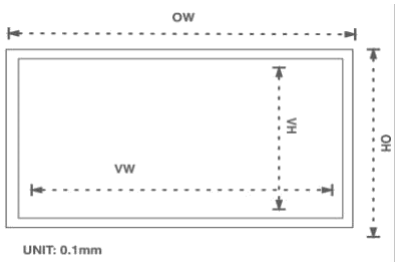
Command:

```
scene change scene1 tx1
```

Response:

```
scene scene1's tx change to tx1
```

2.4.5 scene set bezelgap

Command	scene set *sceneName* bezelgap *vw-nameow oh vw vh*
Response	scene set *sceneName* bezelgap *vw-nameow oh vw vh success*
Description	Sets the bezel compensation parameters for a specific video wall in a scene. • ow&oh= overall width & height of display device including the bezel. vw& vh = screen width &height.  • By default, RX takes ow and vw, oh and vh as being the same, and bezel as zero.

Example:

If you want to set the bezel compensation parameters for video wall vwtest5 in the Scene1:

Command:

```
scene set Scene1 bezelgap vwtest5 1940 1100 1920 1080
```

Response:

```
scene set Scene1 bezelgap vwtest5 1940 1100 1920 1080 success
```

2.4.6 scene set stretch

Command	`scene set sceneName stretch vw-name type`
Response	`scene set sceneName stretch vw-nametype success`
Description	Sets the stretch mode for a specific video wall in a scene. Three options of stretch mode are available as follows; and the default setting is stretch. • **fit:** The picture will scale in proportion; it will be displayed proportionally in maximized state; there may be blank space. • **stretch:** The picture will scale out of proportion; it will be stretched and shown according to the screen resolution; there's no blank space. • **fill:** The picture will scale in proportion to fill the screen; there's no blank space, while part of the picture may not be displayed. **Note:** • **vw-name** is the video wall name. • **Type** is the value of the stretch mode.

Example:

If you want to set the fit mode for the video wall vwtest5 in

Scene1, **Command:**

```
scene set Scene1 stretch vwtest5 fit
```

Response:

```
scene set Scene1 stretch vwtest5 fit success
```

2.4.7 scene connect scenename

Command	`scene connect scenename tx1 tx2 … txnm`
Response	`scene connect scenename tx1 tx2 … txnm success`
Description	Assigns sources to the corresponding RX of a scene in sequence. This action is operated only once and will not be saved in IP controller.

Example:

If you want to assign sources (tx1, tx2, tx3, tx4) to the corresponding RX of scene1 in sequence:

Command:

```
scene connect scene1 tx1 tx2 tx3 tx4
```

Response:

```
scene scene1's tx connect to tx1 tx2 tx3 tx4
```

2.4.8 scene create

Command	`scene create scenesjsonstring`
Response	`scene create success`
Description	Create new scene(s) that defined in json string. Note: *scenesjsonstring* is in standard json string and cannot include characters like carriage

	return or line feed. It can comprise configuration information of multiple scenes and be saved by IP controller.

Example:

Command:

```
scene create
[{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"m":2,"n":2,"name":"scene2-2x
22","rxArray":[[{"aliasName":"visualm7","devName":"IPD5100-341B228010A1","deviceType":"
Receiver","group":[{"name":"group1","sequence":0}],"online":true,"sequence":0,"trueName":
"RX-341B228010A1","txName":"DIPE5100-341B228010BA"},{"aliasName":"visualm12","devName
":"IPD5100-341B228010B0","deviceType":"Receiver","group":[{"name":"undefined","sequence
":0}],"online":true,"sequence":0,"trueName":"RX-341B228010B0","txName":"DIPE5100-341B22801
0BA"}],[{"aliasName":"visualm8","devName":"IPD5100-341B228010AB","deviceType":"Receiver
","group":[{"name":"undefined","sequence":0}],"online":true,"sequence":0,"trueName":"RX-
341B228010AB","txName":"DIPE5100-
341B228010BA"},{"aliasName":"visualm13","devName":"IPD51
00-341B22800BCC","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],
"online":true,"sequence":0,"trueName":"IPD500-341B22800BCC","txName":"DIPE5100-
341B228010
BC"}]],"sceneAutoApply":false,"sequence":0,"txListArray":[[{"devices":[]},{"devices":[]}]
,[{"devices":[]},{"devices":[]}]],"vwConfigList":[{"col_count":2,"mode":"0","name":"vw1",
"oh":0,"ow":0,"pos_col":0,"pos_row":0,"row_count":1,"vh":0,"vw":0}]},{"group":[{"name":"g
roup1","sequence":0}],"layoutseq":0,"m":2,"n":2,"name":"scene2-22","rxArray": [[{"alias
Name":"visualm7","devName":"IPD5100-341B228010A1","deviceType": "Receiver","group”:[{"
name":"undefined","sequence":0}],"online":true,"sequence":0,"trueName":"RX-341B228010A1”,
"txName":"DIPE5100-341B228010BA”}, {"aliasName":"visualm12","devName":"IPD5100-341B22801
0B0”,"deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online":true,"s
equence":0,"trueName":"RX-341B228010B0","txName":"DIPE5100-
341B228010BA”}],[{"aliasName”:
"visualm8","devName":"IPD5100-341B228010AB","deviceType": "Receiver","group":[{"name":"u
ndefined","sequence":0}],"online":true,"sequence":0,"trueName":"RX-341B228010AB”,”txName”
:"DIPE5100-341B228010BA"},{"aliasName":"visualm13","devName":"IPD5100-341B22800BCC","dev
iceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online":true,"sequence":
0,"trueName":"IPD5100-341B22800BCC","txName":"DIPE5100-
341B228010BC"}]],"sceneAutoApply”:
false,"sequence":0,"txListArray":[[{"devices":[]},{"devices":[]}],[{"devices":[]},{"devic
es":[]}]]}]
```

Response:

```
scene create success
```

2.4.9 scene update

Command	`scene update` *`scenesjsonstring`*
Response	`scene update success`
Description	Use the configuration information that defined in json string to update the specified scene. Note: *scenesjsonstring* is in standard json string and cannot include characters like carriage return or line feed. It can comprise configuration information of multiple scenes and be saved by IP controller.

Example:

If you want to update configuration information for the specified scenes:

Command:

```
scene update
[{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"m":2,"n":2,"name":"scene2-
2x22","rxArray":[[{"aliasName":"visualm7","devName":"IPD5100-
341B228010A1","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010A1","txName":"DIPE5100-
341B228010BA"},{"aliasName":"visualm12","devName":"IPD5100-
341B228010B0","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010B0","txName":"DIPE5100-
341B228010BA"}],[{"aliasName":"visualm8","devName":"IPD5100-
341B228010AB","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010AB","txName":"DIPE5100-
341B228010BA"},{"aliasName":"visualm13","devName":"IPD5100-
341B22800BCC","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"IPD5100-341B22800BCC","txName":"DIPE5100-
341B228010BC"}]],"sceneAutoApply":false,"sequence":0,"txListArray":[[{"devices":[]},{"dev
ices":[]}],[{"devices":[]},{"devices":[]}]],"vwConfigList":[{"col_count":2,"mode":"0","na
me":"vw1","oh":0,"ow":0,"pos_col":0,"pos_row":0,"row_count":1,"vh":0,"vw":0}]},{"group":[
{"name":"group1","sequence":0}],"layoutseq":0,"m":2,"n":2,"name":"scene2-
22","rxArray":[[{"aliasName":"visualm7","devName":"IPD5100-
341B228010A1","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010A1","txName":"DIPE5100-
341B228010BA"},{"aliasName":"visualm12","devName":"IPD5100-
341B228010B0","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010B0","txName":"DIPE5100-
341B228010BA"}],[{"aliasName":"visualm8","devName":"IPD5100-
341B228010AB","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"RX-341B228010AB","txName":"DIPE5100-
341B228010BA"},{"aliasName":"visualm13","devName":"IPD5100-
341B22800BCC","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}],"online
":true,"sequence":0,"trueName":"IPD5100-341B22800BCC","txName":"DIPE5100-
341B228010BC"}]],"sceneAutoApply":false,"sequence":0,"txListArray":[[{"devices":[]},{"dev
ices":[]}],[{"devices":[]},{"devices":[]}]]]}
```

Response:

```
scene update success
```

2.4.10 scene modify name

Command	`scene modify name` *`scenename_old1 scenename_new1 scenename_old2 scenename_new2 ...`*
Response	`scene modify name success`
Description	Change the names of specified scenes to new ones. • *scenename_old*: The current name of the scene. • *scenename_new*: The new name of the scene.

Example:

If you want to change the name of the scene from *vw-scene1 to vw-scene2*:

Command:

```
scene modify name vw-scene1 vw-scene2
```

Response:

```
scene modify name vw-scene1 vw-scene2 success
```

2.4.11 scene remove

Command	`scene remove` *`scenename1 scenename2 …`*
Response	`scene remove` *`scenename1 scenename2 …`* `success`
Description	Remove the specified scene(s).

Example:

If you want to remove the scene *vw-scene1*:

Command:

```
scene remove vw-scene1
```

Response:

```
scene remove vw-scene1 success
```

2.4.12 scene group update

Command	`scene group update` *`[json]`*
Response	`scene group or sort update success`
Description	Update the group information of the scenes.

Example:

If you want to update the group information of a scene:

Command:

```
scene group update
[{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"name":"scene2-2x22"}]
```

Response:

```
scene group or sort update success
```

2.5 serial Commands

Command	`serial -b param -r {on|off} -n {on|off}-h {on|off} "command-string"hostname1 hostname2 ...`
Response	`serial command received:` `serial -b param -r {on|off}-n {on|off} -h {on|off} "command-string" hostname1 hostname2 ...`

Description	• Command devices (hostname1, hostname2)to execute port commands. • "And" can not be include in the command string. • ***- b param*** refers to parameters setting of ports that connected to TX/RX (Baud rate, Data bits, Parity, Stop bits). Take -b 115200-8n1 for example. It's a selectable parameter.115200-8n1 is in default. Baud rate can be [150|200|300|600|1200|1800|2400|4800|9600|19200|38400|57600|115200];Data Bits can be [5|6|7|8];Parity can be[o|e|n]; Stop bits can be [1|2]. • ***-r {on|off}*:** Whether to add carriage return at the end of command string. It is a selectable parameter. The default setting is on. • ***-n {on|off}*:** Whether to add new line at the end of command string. It is a selectable parameter. The default setting is off. • ***-h {on|off}*:** When serial commands with -h on, it means that the command-string can be hexadecimal, every 2 characters can be separated by space. For example: AB CD EF 01 23 45 • ***hostname1, hostname2*** refer to destinations.

2.1 "command" Command

Command	`command device-name message-body`
Response	The response may vary based on the specific command.
Description	Sends shell commands to the corresponding TX/RX device(s) via the IP control box. There will not be response in the following two cases: 1) The API command itself doesn't return with any response; 2) The API command is an asynchronous message, the third party controller may require following processing through some commands (e.g. "notify", "get") to get message. **Note**: The "command" command may not be applicable to all commands, please test before using it.

Example:

Command:

```
commandDIPD-5100-341B22800BCD hostname
```

Response:

```
IPD5100-341B22800BCD
```

Command:

```
commandDIPD-5100-341B22800304 gbparam s nameDIPD-5100-1; gbparam g name
```

Response:

```
IPD5100-1
```

2.2 notify Commands

notify commands are positively sent to a third-party control device such as a PC from IP controller. Those command for notify are automatically sent to the third-party controller or application. It shows some status changing in the system, and third party controller can capture this information from the session, and offer it to the application layer. Other commands are sent from a third party control device, then IP controller executes them and gives response to the control device. The commands in this section have no requests and responses.

2.2.1 notify endpoint

Command	`notify endpoint {+|-} hostname1 hostname2... {-|+} hostnameM hostnameN...`
Description	Positively informs a third-party control device that devices just got online or offline when devices' online or offline status changes. **Note:** "+" indicates that devices just got online. "–" indicates that devices just got offline.

Example:

IP controller informs a third party control device that DIPE5100-341B22800BB0 just got online.

```
notify endpoint + DIPE5100-341B22800BB0
```

2.2.2 notify serialinfo

Command	`notify serialinfo hostname {hex|ascii} infolen:\r\ninfo\r\n`
Description	Positively informs a third party control device about the data received in a device's serial port. **Note:** • **hostname** 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. • **infolen** is the length of **info**. Unit is byte.**info** is the actual data received. For ASCII data, **infolen** is the number of actual data bytes received. For hexadecimal data, **(infolen+1)/3** is the number of actual data bytes received. • **\r** and **\n** are escape characters, meaning a carriage return and a line feed respectively.

Example1:

DIPE5100-341B228007CB's 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"):

```
notify serialinfo DIPE5100-341B228007CB hex 56:
68 65 6C 11 6C 6F 11 22 33 44 00 55 66 77 99 AA CC DD FF
```

Example2:

IPD5100-341B22800BCA's serial port receives five characters "12345":

```
notify serialinfoDIPD-5100-341B22800BCA ascii 5:
12345
```

2.2.3 notify irinfo

Command	`notify irinfo dev "IRDATA"`
Description	The IP controller positively informs a third party control device about the IR data of a device.

Example:

IP controller informs a third party control device about the DVD's IR data as follows.

```
notify irinfo DVD "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018
0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018
0030 0018 0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018
0030 01FE"
```

2.2.4 notify cecinfo (for DIPX-5100)

Command	`notify cecinfo dev "CECDATA"`
Description	The IP controller positively informs a third-party control device about the cec information of a device.

Example:

IP controller informs a third party control device about a device's cec information as follows.

```
notify cecinfo DVD "ff36"
```

2.2.5 notify video (for DIPX-5100)

Command	`notify video {lost|found} tx/rx [(VideoSourceName)]`
Description	Positively informs the third party control device about the video signal lost or restored status. **Note:** • **VideoSourceName** descript the name with source. It's only useful for rx for now

Example:

DIPE5100-341B22800BB0 got the video lost from the system, then the IP controller will offer the notification like under:

```
notify video lost DIPE5100-341B22800BB0
```

2.2.6 notify update (for IPX6000)

Command	`notify update DEV status`
Description	Positively informs the third-party control device about the update status of a device. **Note:** There're three kinds of status: • **success:** indicates the device updates successfully. • **failed:** indicates the device fails to update.

	• **processing:** indicates that the device is updating, a percentage of the update progress will be shown as well.

IP controller informs a third-party control device about a device's update status.

Example 1:

```
notify update RX1 success
```

Example 2:

```
notify update RX1 processing 56
```

2.3 mscene Commands (for IPD6000)

To use **mscene** (Multiview scene) commands, the layouts for the applicable devices must be pre-set by configurator.

2.3.1 mscene get

Command	mscene get *[mrx1 mrx2 … mrxn]*
Response	mscene list:\r\n *mrx1 layout11 layout12 …layout1m*\r\n *mrx2 layout21 layout22 … layout2m*\r\n\ ... *mrxn layoutn1 layoutn2 … layoutnm*\r\n \r\n
Description	Obtain the multiview layout names of the specified or all applicable devices. If no device is specified, the controller will return with all Multiview layout names of devices' that support Multiview scene in the system.

Example:

If you want to obtain all multiview layout names of the devices' that support Multiview scene:

Command:

```
mscene get
```

Response:

```
mscene list:\r\n
IPD6000-1 1-1 1-3 1-4 2-2 3-1\r\n
IPD6000-2 2-2 3-3\r\n
IPD6000-3 1-1 1-3 1-4\r\n
\r\n
```

2.3.2 mscene getjson

Command	mscene getjson *[mrx1 mrx2 … mrxn]*
Response	mscene json string:\r\n *jsonstring* \r\n

Description	Obtain json information of all applicable or specified devices' Multiview scenes.

Example:

If you want to obtain json information of the IPD6000-1's Multiview scene:

Command:

```
mscene getjson IPD6000-1
```

Response:

```
mscene json string:

[
    {
        "group" : [
            {
                "name" : "group2",
                "sequence" : 2
            }
        ],
        "hsize" : 1920,
        "aliasName" : "IPD6000-1",
        "trueName" : "IPD6000-341B228000BB",
        "sequence" : 0,
        "layouts" : [
            {
                "name" : "2-1",
                "layoutseq" : 1,
                "windows" : [
                    {
                        "hsize" : 960,
                        "hstart" : 0,
                        "mode" : "fit",
                        "name" : "1",
                        "tx" : "DIPE5100-341B228010BD",
                        "vsize" : 540,
                        "vstart" : 270
                    },
                    {
                        "hsize" : 960,
                        "hstart" : 960,
                        "mode" : "fit",
```

```
                        "name" : "2",
                        "tx" : "DIPE5100-341B228010BE",
                        "vsize" : 540,
                        "vstart" : 270
                    }
                ]
            },
            {
                "name" : "1-1",
                "layoutseq" : 2,
                "windows" : [
                    {
                        "hsize" : 1920,
                        "hstart" : 0,
                        "mode" : "fit",
                        "name" : "1",
                        "tx" : "DIPE5100-341B228010BD",
                        "vsize" : 1080,
                        "vstart" : 0
                    }
                ]
            }
        ],
        "vsize" : 1080
    }
]
```


2.3.3 mscene active

Command	mscene active *mrxname layoutname*
Response	mscene active *mrxname layoutname {success|failure}*
Description	Apply a certain multiview layout for the specified device.

Example:

If you want to apply the 1-1 layout for IPD6000-1:

Command:

```
mscene active IPD6000-1 1-1
```

Response:

```
mscene active IPD6000-1 1-1 success
```

2.3.4 mscene change mrxname layoutname

Command	`mscene change mrxname layoutname window1 txname1 window2 txname2…`
Response	`mscene change mrxname layoutname window1 txname1 window2 txname2… {success|failure}`
Description	Assign specified TX(s) to the certain window(s) in the certain multiview layout for all applicable or specified device(s).

Example:

If you want to assign the tx "DIPE6000-341B22FFFFC4" to the window1 of the IPD6000-1's 1-1 layout:

Command:

```
mscene change IPD6000-1 1-1 1 DIPE6000-341B22FFFFC4
```

Response:

```
mscene change IPD6000-1 1-1 1 DIPE6000-341B22FFFFC4 success
```

2.3.5 mscene changeall mrxname layoutname

Command	`mscene changeall mrxname layoutname txname`
Response	`mscene changeall mrxname layoutname txname {success|failure}`
Description	Assign a specified TX to all windows of the certain Multiview layout for the certain device.

Example:

If you want to assign the tx "TX1" to all windows of the layout 1-1 for IPD6000-1:

Command:

```
mscene changeall IPD6000-1 1-1 tx1
```

Response:

```
mscene changeall IPD6000-1 1-1 tx1 success
```

2.3.6 mscene set audio

Command	`mscene set audio mrxname layoutname{follow/window/separate} [{windowname/tx}]`
Response	`mscene set audio mrxname layoutname {follow/window/separate} [{windowname/tx}] {success/failure}`
Description	Configure the audio mode for the specified device's layout. *{follow/window/separate}* is the audio mode of the layout. • In follow mode, the audio in the layout follows the video if the layout is set as single view, and there will be no audio output when the layout is set as multiview. • In window mode, the audio in the layout will follow the video source in the specified window.

	• In separate mode, the audio source in the layout is specified by the parameter "tx". The default mode of layout is *follow*. Attention: this command just changes the audio mode setting of the layout, and the setting will take affect when you execute the command “mscene active”.

Example:

If you want to configure the audio in the 2-1 layout of IPD6000-1 to window mode:

Command:

```
mscene set audio IPD6000-1 2-1 window 1
```

Response:

```
mscene set audio IPD6000-1 2-1 window 1 success
```

2.3.7 mscene changeaudio

Command	`mscene changeaudio` *`mrxname {follow/separate} [tx]`*
Response	`mscene changeaudio` *`mrxname {follow/separate} [tx] {success/failure}`*
Description	Change the audio mode for all layouts of the specified MRX, and change the audio mode of the MRX at the same time. *`{follow/separate}`* *i*s the audio mode of the layout, that is follow or separate. In follow mode, the audio in the layout follows the video if the layout is single view layout, and no audio out when layout is multiview layout. In separate mode, the audio source in the layout is specified by the parameter "tx". The default mode of layout is *follow*.

Example:

If you want to change the audio mode for all layouts of IPD6000-1 to separate mode:

Command:

```
mscene changeaudio IPD6000-1 follow
```

Response:

```
mscene changeaudio IPD6000-1 follow success
```

2.3.8 mscene create

Command	`mscene create []`
Response	`mscene create success` *`\r\n`*
Description	To create and define multiview scenes for the specified devices using json information.

Example:

Command:

```
mscene create
[
    {
        "group" : [
            {
                "name" : "group2",
                "sequence" : 2
            }
        ],
        "hsize" : 1920,
        "aliasName" : "IPD6000-1",
        "trueName" : "IPD6000-341B228000BB",
        "sequence" : 0,
        "layouts" : [
            {
                "audio" : {
                    "mode" : "follow",
                    "source" : ""
                },
                "name" : "2-1",
                "layoutseq" : 1,
                "windows" : [
                    {
                        "hsize" : 960,
                        "hstart" : 0,
                        "mode" : "fit",
                        "name" : "1",
                        "tx" : "DIPE6000-341B228010BD",
                        "vsize" : 540,
                        "vstart" : 270
                    },
                    {
                        "hsize" : 960,
                        "hstart" : 960,
                        "mode" : "fit",
                        "name" : "2",
                        "tx" : "DIPE6000-341B228010BE",
```

```
                    "vsize" : 540,
                    "vstart" : 270
                }
            ]
        },
        {
            "name" : "1-1",
            "layoutseq" : 2,
            "windows" : [
                {
                    "hsize" : 1920,
                    "hstart" : 0,
                    "mode" : "fit",
                    "name" : "1",
                    "tx" : "DIPE6000-341B228010BD",
                    "vsize" : 1080,
                    "vstart" : 0
                }
            ]
        }
    ],
    "vsize" : 1080
  }
]
```

Response:

```
mscene create success
```

2.3.9 mscene group update

Command	mscene group update *[json]*
Response	mscene group or sort update success
Description	To change grouping or sequence for the specified devices.

Example:

Command:

```
mscene group update [{"group":[{"name":"group1","sequence":0}]," sequence ":0,"
trueName ":"IPD6000-341B2280FF02"}]
```

Response:

```
mscene group or sort update success
```

2.3.10 Structure of the mscene

Name	Description	Remark
aliasName	Alias name of the mscene to the RX device	
trueName	True name of the mscene to the RX device	
hsize	The horizontal physical pixels of the display screen at RX (default value is 1920)	
vsize	The vertical physical pixels of the display screen at RX (default value 1080)	
sequence	The sequence number of the mscene in the group	
group.name	Group name of the RX	
group.sequence	The position number of the RX device in a group set	
layouts.i.name	Layout's name	
layouts.i.layoutseq	Location sequence on the layout for the RX	
layouts.i.windows.j.name	Window's name for No j window	
layouts.i.windows.j.layerseq	The sequence of the window in the layout.	The sequence of the layout is from the bottom to the top. Number begins from 1. But don't need to concern for now
layouts.i.windows.j.hstart	Start horizontal pixel to the window	Refer to the whole RX video input
layouts.i.windows.j.vstart	Start vertical pixel to the window	Refer to the whole RX video input
layouts.i.windows.j.hsize	Horizontal pixels to the window	Refer to the whole RX video output
layouts.i.windows.j.vsize	Vertical pixels to the window	Refer to the whole RX video output
layouts.i.windows.j.tx	Default display source name for the window	TX with the true name
layouts.i.windows.j.mode	Display mode on the window	

2.4 wscene2 Commands (for DIPD-5100)

The "video wall within a wall" application, also referred to "Picture in Picture"(PIP). Small video walls can be overlapped on the top of any larger video wall. It is often to be used in situation with multiple scenes. As shown in figures below:

2.4.1 wscene2 get

Command	`wscene2 get`
Response	`wscene2 list:` `wscene1 wscene2`
Description	Obtain all names of windowing scenes.

Example:

If you want to obtain allz names of the windowing scenes:

Command:

```
wscene2 get
```

Response:

```
wscene2 list:
2x2-fullscreen1 wall2-fullscreen
```

2.4.2 wscene2 getjson

Command	wscene2 getjson wscenename
Response	wscene2 json string:\r\n *jsonstring*
Description	Obtain descriptive information of all or specified windowing scenes in json string.

Example:

If you want to obtain information of all scenes in json string:

Command:

```
wscene2 getjson
```

Response:

```
wscene2 json string:
[
    {
        "group" : [
            {
                "name" : "group2",
                "sequence" : 3
            }
        ],
        "hsize" : 1920,
        "layoutseq" : 3,
        "m" : 2,
        "maxlayerinrx" : 10,
        "n" : 2,
        "name" : "2x2-fullscreen1",
        "rxArray" : [
            [
                {
                    "aliasName" : "IPD5100-5flash",
                    "deviceType" : "Receiver",
                    "group" : [
                        {
                            "name" : "group2",
                            "sequence" : 3
                        }
                    ],
```

```
            "online" : false,
            "sequence" : 3,
            "trueName" : "IPD5100-341B22802B8F",
            "txName" : ""
        },
        {
            "aliasName" : "IPD5100-2",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "group2",
                    "sequence" : 3
                }
            ],
            "online" : false,
            "sequence" : 2,
            "trueName" : "IPD5100-341B2280108B",
            "txName" : ""
        }
    ],
    [
        {
            "aliasName" : "IPD5100-6flash",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "group2",
                    "sequence" : 3
                }
            ],
            "online" : false,
            "sequence" : 4,
            "trueName" : "IPD5100-341B22802B90",
            "txName" : ""
        },
        {
            "aliasName" : "IPD5100-3",
```

```
                "deviceType" : "Receiver",
                "group" : [
                    {
                        "name" : "group2",
                        "sequence" : 3
                    }
                ],
                "online" : false,
                "sequence" : 1,
                "trueName" : "IPD5100-341B22801070",
                "txName" : ""
            }
        ]
    ],
    "sequence" : 1,
    "vsize" : 1080
},
{
    "group" : [
        {
            "name" : "group1",
            "sequence" : 2
        }
    ],
    "hsize" : 1920,
    "layoutseq" : 2,
    "m" : 1,
    "maxlayerinrx" : 10,
    "n" : 1,
    "name" : "wall2-fullscreen",
    "rxArray" : [
        [
            {
                "aliasName" : "IPD5100-4",
                "deviceType" : "Receiver",
                "group" : [
                    {
```

```
                        "name" : "group1",
                        "sequence" : 2
                    }
                ],
                "online" : false,
                "sequence" : 1,
                "trueName" : "IPD5100-341B22802B92",
                "txName" : ""
            }
        ]
    ],
    "sequence" : 1,
    "vsize" : 1080
}
```


2.4.3 wscene2 active

Command	`wscene2 active` *`wscenename`*
Response	`wscene2 active` *`wscenename {success|failure}`*
Description	Apply the specified windowing scene.

Example:

If you want to apply the windowing scene "*2x2-fullscreen1"*:

Command:

```
wscene2 active 2x2-fullscreen1
```

Response:

```
wscene2 active 2x2-fullscreen1 success
```

2.4.4 wscene2 create

Command	`wscene2 create` *`wscenesjsonstring`*
Response	`wscene2 create success`
Description	Create a windowing scene that defined by *wscenesjsonstring* in server.

Example:

If you want to create a windowing scene defined in json string for target devices:

Command:


```
wscene2 create
[{"group":[{"name":"group11","sequence":2}],"hsize":1920,"layoutseq":4,"m":2,"max
layerinrx":10,"n":2,"name":"vw1-
```

layout4","rxArray":[[{"aliasName":"VisualM01","audioTxName":"null","devName":"IPD
5100-
341B22800BCD","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B22800BCD","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM04","audioTxName":"null","devName":"IPD5100-
341B2280109F","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":3,"trueName":"IPD5100-
341B2280109F","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"VisualM02","audioTxName":"null","devName":"IPD5100-
341B228010A1","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":4,"trueName":"IPD5100-
341B228010A1","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM05","audioTxName":"null","devName":"IPD5100-
341B22800BD3","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B22800BD3","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":7,"vsize":1080,"windows":[{"hsize":1,"hstart":1,"layerseq":1,"name":"iosfullsc
reen5062941010","tx":"DIPE5100-
341B228010BA","vsize":1,"vstart":1}]},{"group":[{"name":"group11","sequence":2}],
"hsize":1920,"layoutseq":6,"m":2,"maxlayerinrx":10,"n":2,"name":"vw1-
layout6","rxArray":[[{"aliasName":"VisualM01","audioTxName":"null","devName":"IPD
5100-
341B22800BCD","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B22800BCD","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM04","audioTxName":"null","devName":"IPD5100-
341B2280109F","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":3,"trueName":"IPD5100-
341B2280109F","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"VisualM02","audioTxName":"null","devName":"IPD5100-
341B228010A1","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":4,"trueName":"IPD5100-
341B228010A1","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM05","audioTxName":"null","devName":"IPD5100-
341B22800BD3","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B22800BD3","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":7,"vsize":1080,"windows":[{"hsize":2,"hstart":0,"layerseq":1,"name":"iosfullsc
reen4011951010","tx":"DIPE5100-
341B228010BA","vsize":2,"vstart":0}]},{"group":[{"name":"group11","sequence":2}],
"hsize":1920,"layoutseq":5,"m":1,"maxlayerinrx":10,"n":2,"name":"vw2-
layout5","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD
5100-
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B228010AE","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"visualM10","audioTxName":"null","devName":"IPD5100-

341B228010A4","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B228010A4","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":9,"vsize":1080},{"group":[{"name":"group11","sequence":2}],"hsize":1920,"layou
tseq":6,"m":1,"maxlayerinrx":10,"n":2,"name":"vw2-
layout6","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD
5100-
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B228010AE","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"visualM10","audioTxName":"null","devName":"IPD5100-
341B228010A4","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B228010A4","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":9,"vsize":1080}],"wscenelist":[{"autoapply":0,"group":[{"name":"group11","sequ
ence":2}],"hsize":1920,"layoutseq":5,"m":2,"maxlayerinrx":4,"n":2,"name":"vw1-
layout5","rxArray":[[{"aliasName":"VisualM01","audioTxName":"null","devName":"IPD
5100-
341B22800BCD","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B22800BCD","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM04","audioTxName":"null","devName":"IPD5100-
341B2280109F","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":3,"trueName":"IPD5100-
341B2280109F","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"VisualM02","audioTxName":"null","devName":"IPD5100-
341B228010A1","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":4,"trueName":"IPD5100-
341B228010A1","txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasNam
e":"VisualM05","audioTxName":"null","devName":"IPD5100-
341B22800BD3","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B22800BD3","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":7,"vsize":1080},{"autoapply":0,"group":[{"name":"group11","sequence":2}],"hsiz
e":1920,"layoutseq":3,"m":1,"maxlayerinrx":4,"n":2,"name":"vw2-
layout3","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD
5100-
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B228010AE","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"visualM10","audioTxName":"null","devName":"IPD5100-
341B228010A4","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B228010A4","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":9,"vsize":1080},{"autoapply":0,"group":[{"name":"group11","sequence":2}],"hsiz
e":1920,"layoutseq":4,"m":1,"maxlayerinrx":4,"n":2,"name":"vw2-
layout4","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD
5100-

```
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-
341B228010AE","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasN
ame":"visualM10","audioTxName":"null","devName":"IPD5100-
341B228010A4","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"
online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-
341B228010A4","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequenc
e":9,"visualscreens":[{"hsize":1920,"hstart":-
20,"layerseq":1,"name":"iosscreen1792761008","tx":"DIPE5100-
341B228010BA","vsize":1080,"vstart":603}],"vsize":1080}]
```

Response:

```
wscene2 create success
```

2.4.5 wscene2 update

Command	wscene2 update *wscene2sjsonstring*
Response	wscene2 update success
Description	Use the configuration information of *wscene2sjsonstring* to update the corresponding windowing scene existed in server and apply it to the specified device.

Example:

If you want to apply the configuration information of *wscene2sjsonstring* to a device:

Command:

```
wscene2 update [{"group":[{"name":"group11", "sequence":2}], "hsize":1920, "layoutseq":
4,"m":2,"maxlayerinrx":10,"n":2,"name":"vw1-layout4", "rxArray":[[{"aliasName":"Visu
alM01","audioTxName":"null","devName":"IPD5100-341B22800BCD","deviceType":"Receiver",
"group":[{"name":"group11","sequence":2}],"online":true,"rxstatus":1,"sequence":1,"trueN
a me":"IPD5100-341B22800BCD","txName":"null", "usbTxName":"null", "videoTxName": "null"}
,{"aliasName":"VisualM04","audioTxName":"null","devName":"IPD5100-341B2280109F"
,"deviceType":"Receiver", "group":[{"name":"group11", "sequence":2}],"online":true,
"rxstatus":1,"sequence":3,"trueName":"IPD5100-341B2280109F", "txName":
"null","usbTxName":"null","videoTxName":"null"}],[{"aliasName":"VisualM02","audioTxName"
: "null","devName":"IPD5100-341B228010A1","deviceType": "Receiver", "group": [{"name":
"group11","sequence":2}], "online":true,"rxstatus":1,"sequence":4, "trueName" :"IPD5100-
341B228010A1","txName":"null","usbTxName":"null", "videoTxName":"null"},
{"aliasName":"VisualM05","audioTxName":"null","devName":"IPD5100-341B22800BD3",
"deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"online":true,"rxstatu
s ":1,"sequence":2,"trueName":"IPD5100-341B22800BD3","txName":"null", "usbTxName":
"null","videoTxName":"null"}]],"sequence":7,"vsize":1080,"windows":[{"hsize":1,"hstart":
1 ,"layerseq":1,"name":"iosfullscreen5062941010","tx":"DIPE5100-341B228010BA",
"vsize":1, "vstart":1}]}, {"group":[{"name":"group11","sequence":2}],"hsize":1920,
"layoutseq":6, "m" :2,"maxlayerinrx":10,"n":2,"name":"vw1-layout6", "rxArray":
[[{"aliasName": "VisualM0 1","audioTxName":"null","devName":"IPD5100-341B22800BCD",
"deviceType":"Receiver", "group":[{"name":"group11","sequence":2}],"online":true,
"rxstatus":1, "sequence":1, "trueName":"IPD5100-341B22800BCD", "txName":"null",
"usbTxName": "null", "videoTxName"
:"null"},{"aliasName":"VisualM04","audioTxName":"null","devName":"IPD5100-341B2280109F",
"deviceType":"Receiver", "group":[{"name":"group11","sequence":2}],"online":true,
```

```
"rxstatus":1,"sequence":3, "trueName":"IPD5100-341B2280109F", "txName":"null", "usbTxNa
me":"null","videoTxName":"null"}],[{"aliasName":"VisualM02","audioTxName":"null","devNam
e ":"IPD5100-341B228010A1","deviceType":"Receiver","group":[{"name": "group11",sequence
":2}],"online":true,"rxstatus":1,"sequence":4,"trueName":"IPD5100-341B228010A1"
,"txName":"null","usbTxName":"null","videoTxName":"null"},{"aliasName":"VisualM05","audi
o TxName":"null","devName":"IPD5100-341B22800BD3","deviceType":"Receiver", "group"
:[{"name":"group11","sequence":2}],"online":true,"rxstatus":1,"sequence":2,"trueName":"I
P D5100-
341B22800BD3","txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequence"
:7,"vsize":1080,"windows":[{"hsize":2,"hstart":0,"layerseq":1,"name":"iosfullscreen40119
5 1010","tx":"DIPE5100-341B228010BA","vsize":2,"vstart":0}]},{"group":
[{"name":"group11",
"sequence":2}],"hsize":1920,"layoutseq":5,"m":1,"maxlayerinrx":10,"n":2,"name":"vw2-
layout5","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD5100-
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"online"
: true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-341B228010AE","txName":"null",
"usbTxName":"null","videoTxName":"null"}],[{"aliasName":"visualM10","audioTxName":"null"
, "devName":"IPD5100-341B228010A4","deviceType": "Receiver","group": [{"name":"group11",
"sequence":2}],"online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-341B228010A4"
,"txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequence":9,"vsize":1080},{
"
group":[{"name":"group11","sequence":2}],"hsize":1920,"layoutseq":6,"m":1,"maxlayerinrx"
: 10,"n":2,"name":"vw2-layout6","rxArray":[[{"aliasName":"VisualM09",
"audioTxName":"null", "devName":"IPD5100-341B228010AE","deviceType":
"Receiver","group":[{"name":"group11",
"sequence":2}],"online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-341B228010AE"
,"txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasName":"visualM10","au
d ioTxName":"null","devName":"IPD5100-341B228010A4","deviceType":"Receiver","group":
[{"name":"group11","sequence":2}],"online":true,"rxstatus":1,"sequence": 2,"trueName"
:"IPD5100-341B228010A4","txName":"null", "usbTxName":"null", "videoTxName":"null"}]],
"sequence":9,"vsize":1080}],"wscenelist":[{"autoapply":0,"group":[{"name":"group11","seq
u ence":2}],"hsize":1920,"layoutseq":5,"m":2,"maxlayerinrx":4,"n":2,"name":"vw1-
layout5","rxArray":[[{"aliasName":"VisualM01","audioTxName":"null","devName":"IPD5100-
341B22800BCD","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"online"
: true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-341B22800BCD", "txName":"null",
"usbTxName":"null","videoTxName":"null"},{"aliasName":"VisualM04","audioTxName":"null","
d evName":"IPD5100-341B2280109F","deviceType": "Receiver","group":[{"name":"group11",
"sequence":2}],"online":true,"rxstatus":1,"sequence":3,"trueName":"IPD5100-341B2280109F
","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasName":"VisualM02","a
u dioTxName":"null","devName":"IPD5100-341B228010A1","deviceType": "Receiver","group"
:[{"name":"group11","sequence":2}],"online":true,"rxstatus":1,"sequence":4,"trueName":"I
P D5100-341B228010A1","txName":"null","usbTxName": "null","videoTxName":"null"},{"alias
Name":"VisualM05","audioTxName":"null","devName":"IPD5100-341B22800BD3", "deviceType":
"Receiver","group":[{"name":"group11","sequence":2}], "online":true,"rxstatus":1,
"sequence":2,"trueName":"IPD5100-341B22800BD3","txName":"null","usbTxName":"null","vi
deoTxName":"null"}]],"sequence":7,"vsize":1080},{"autoapply":0,"group":[{"name":"group11
" ,"sequence":2}],"hsize":1920,"layoutseq":3,"m":1,"maxlayerinrx":4,"n":2,"name":"vw2-
layout3","rxArray":[[{"aliasName":"VisualM09","audioTxName":"null","devName":"IPD5100-
341B228010AE","deviceType":"Receiver","group":[{"name":"group11","sequence":2}],"online"
: true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-341B228010AE",txName":"null",
usbTxName":"null","videoTxName":"null"}],[{"aliasName":"visualM10","audioTxName":"null",
" devName":"IPD5100-341B228010A4","deviceType": Receiver","group":[{"name":"group11","s
```

quence":2}],"online":true,"rxstatus":1,"sequence":2,"trueName":"IPD5100-341B228010A4",
txName":"null","usbTxName":"null","videoTxName":"null"}]],"sequence":9,"vsize":1080},{"au
toapply":0,"group":[{"name":"group11","sequence":2}],"hsize":1920,"layoutseq":4,"m":1,"ma
xlayerinrx":4,"n":2,"name":"vw2-layout4","rxArray" [[{"aliasName":"VisualM09" "audioTxNam
""null","devName":"IPD5100-341B228010AE", deviceType":"Receiver", group":[{"name": group1
1","sequence":2}],"online":true,"rxstatus":1,"sequence":1,"trueName":"IPD5100-341B228010A
E","txName":"null","usbTxName":"null","videoTxName":"null"}],[{"aliasName":"visualM10","a
udioTxName":"null","devName":"IPD5100-341B228010A4","deviceType": "Receiver","group"
:[{"name":"group11","sequence":2}],"online":true,"rxstatus":1,"sequence":2,"trueName":"IP D5100-
341B228010A4","txName":"null","usbTxName":"null","videoTxName" :"null"}]], "sequenc
e":9,"visualscreens":[{"hsize":1920,"hstart":-20,"layerseq":1,"name":"iosscreen1792761008 ","tx":"DIPE5100-
341B228010BA", "vsize":1080,"vstart":603}],"vsize":1080}]

Response:

```
wscene2 update success
```

Note:

1. wscene2sjsonstring must not contain any line breaks.
2. This command will update the configuration information of *wscene2sjsonstring* in the IP controller and apply the updated scene to corresponding device.

2.4.6 wscene2 modify name

Command	`wscene2 modify name` *`scenename_old scenename_new`*
Response	`wscene2 modify name` *`scenename_old scenename_new`* `success`
Description	Change the name of a windowing scene to a new one. • *scenename_old*: The current name of the scene. • *scenename_new*: The new name of the scene.

Example:

If you want to change the name of the windowing scene from *vw-scene1 to vw-scene2*:

Command:

```
wscene2 modify name vw-scene1 vw-scene2
```

Response:

```
wscene2 modify name vw-scene1 vw-scene2 success
```

2.4.7 wscene2 remove

Command	`wscene2 remove` *`scenename1 scenename2 ..`*
Response	`wscene2 remove` *`scenename1 scenename2 ..`* `success`
Description	Remove the specified scene(s).

Example:

If you want to remove the scene *vw-scene1*:

Command:

```
wscene2 remove vw-scene1
```

Response:

```
wscene2 remove vw-scene1 success
```

2.4.8 wscene2 group update

Command	`wscene2 group update [json]`
Response	`wscene2 group or sort update success`
Description	Update the group information of the scenes.

Example:

If you want to update the group information of a scene:

Command:

```
wscene2 group update
[{"group":[{"name":"group1","sequence":2}],"layoutseq":0,"name":"scene2-2x22"}]
```

Response:

```
wscene2 group or sort update success
```

Note: The sequence of the group must start from “2”.

2.4.9 wscene2 window open

Command	`wscene2 window open scenename windowname hstart vstart hsize vsize txname`
Response	`wscene2 window open scenename windowname hstart vstart hsize vsize txname {success|failure}`
Description	Create a window “*windowname*” under the “*scenename*” scene that is located by the parameters of *hstart*, *vstart*, *hsize*, *vsize*. • *windowname*: The name of the window to be created. • *scenename:* The name of the full screen windowing scene. • *hstart*: The abscissa of the vertex at the upper left corner of the window. • *vstart*: The ordinate of the vertex at the upper left corner of the window. • *hsize*: The width of the window. • *vsize*: The height of the window. • *txname:* The encoder that is to be assigned to the window. Note: (1) The name of the window “*windowname*” should be unique in the windowing scene “*scenename*”, or you will get a response with “failure”. (2) The newly created window is on the top of all windows by default.

Example:

If you want to create a 2x2 window "*window1*" in the scene "*2x2-fullscreen1*" with its vertex at the upper left corner defined as (0,0):

Command:

```
wscene2 window open 2x2-fullscreen1 window1 0 0 2 2 Encoder1
```

Response:

```
wscene2 window open 2x2-fullscreen1 window1 0 0 2 2 Encoder1 success
```

2.4.10 wscene2 window close

Command	wscene2 window close *scenename windowname*
Response	wscene2 window close *scenename windowname {success|failure}*
Description	Close the specified window in the corresponding windowing scene.

Example:

If you want to close window1 in the scene "2x2-fullscreen1":

Command:

```
wscene2 window close 2x2-fullscreen1 window1
```

Response:

```
wscene2 window close 2x2-fullscreen1 window1 success
```

2.4.11 wscene2 window adjust

Command	wscene2 window adjust *scenename windowname hstart vstart hsize vsize*
Response	wscene2 window adjust *scenename windowname hstart vstart hsize vsize {success|failure}*
Description	Adjust the coordinate parameters of the specified window in the corresponding windowing scene. • *hstart*: The abscissa of the vertex at the upper left corner of the window. • *vstart*: The ordinate of the vertex at the upper left corner of the window. • *hsize*: The width of the window. • *vsize*: The height of the window.

Example:

If you want to adjust *window1* in *2x2-fullscreen1* so that its vertex at the upper left corner is defined as (0,0), width is 1 and height is 2:

Command:

```
wscene2 window adjust 2x2-fullscreen1 window1 0 0 1 2
```

Response:

```
wscene2 window adjust 2x2-fullscreen1 window1 0 0 1 2 success
```

2.4.12 wscene2 window move

Command	`wscene2 window move scenename windowname {up|down|top|bottom}`
Response	`wscene2 window move scenename windowname {up|down|top|bottom} {success|failure}`
Description	Arrange the layer order for the specified window in the corresponding windowing scene. • up: Move the layer forward. • down: Move the layer backward. • top: Move the layer to the top. • bottom: Move the layer to the bottom.

Example:

If you want to move the window1 in 2x2-fullscreen1 in front of all other windows in its layer:

Command:

```
wscene2 window move 2x2-fullscreen1 window1 top
```

Response:

```
wscene2 window move2x2-fullscreen1 window1 top success
```

2.4.13 wscene2 window change

Command	`wscene2 window change scenename windowname1 txname1`
Response	`wscene2 window change scenename windowname1 txname1 … {success|failure}`
Description	Assign sources to the specified windows in the corresponding windowing scene.

Example:

If you want to assign video sources to window1 and window2 respectively in the scene of 2x2-fullscreen1:

Command:

```
wscene2 window change 2x2-fullscreen1 window1 tx1 window2 tx2
```

Response:

```
wscene2 window change 2x2-fullscreen1 window1 tx1 window2 tx2 success
```

2.4.14 wscene2 window changeall

Command	`wscene2 window changeall scenename txname`
Response	`wscene2 window changeall scenename txname {success|failure}`
Description	Assign a source to all windows in the corresponding windowing scene.

Example:

If you want to assign a video source (tx) to all windows in the scene of 2x2-fullscreen1:

Command:

```
wscene2 window changeall 2x2-fullscreen1 tx1
```

Response:

```
wscene2 window changeall 2x2-fullscreen1 tx1 success
```

2.4.15 wscene2 window changetx wscenename

Command	wscene2 window changetx *scenename windowname1 txname1 windowname2 txname2…*
Response	wscene2 window changetx *scenename windowname1 txname1 windowname2 txname2… success*
Description	Assign a source to a window one by one in the corresponding windowing scene. This difference between this command and "wscene2 window change" is that this command is used to assign a video source to each rx in the scene while the latter is used to assign video sources to the specified RXs only.

Example:

If you want to assign encoder 1 and encoder 2 to window 1 and window 2 respectively in the scene of 2x2-fullscreen1:

Command:

```
wscene2 window changetx 2x2-fullscreen1 window1 Encoder1 window2 Encoder2
```

Response:

```
wscene2 window changetx 2x2-fullscreen1 window1 Encoder1 window2 Encoder2 success
```

2.4.16 Structure of the wscene2

Name	Description	Remark
name	The name of wscene2.	
n	The row number of Physical screen.	
m	The column number of Physical screen.	
hsize	Horizontalpixel of each physical screen.	
vsize	Verticalpixel of each physical screen.	
sequence	The sequence of wscene2 video wall in the group.	
layoutseq	The sequence of wscene2 layout in the video wall	
maxlayerinrx	Maximum layer in physical RX.	
group.name	The name for the group that wscene2 is in.	
group.sequence	The sequence of group that wscene2 is in.	
rxArray.i.aliasName	The alias name of RX in position i.	
rxArray.i.trueName	The true name of RX in position i.	
rxArray.i.online	Whether RX is online in position i.	
windows.i.name	The name for window No. i.	
windows.i.layerseq	The layer sequence for window No. i.	The layer sequence is from bottom to top started from 1.
windows.i.hstart	The horizontal axis of upper-left corner of window No. i in the video wall.	Use the whole video wall as reference and device as unit, increasing from left to right with 0 as start.
windows.i.vstart	The vertical axis of upper-left corner of window No. i in the video wall.	Use the whole video wall as reference and device as unit,

Name	Description	Remark
		increasing from top to bottom with 0 as start.
windows.i.hsize	The length in horizontal of window No. i in the video wall.	Use the whole video wall as reference and device as unit.
windows.i.vsize	The length in vertical of window No. i in the video wall.	Use the whole video wall as reference and device as unit.
windows.i.tx	The default source of window No. i.	TX is the true name of the device.

2.5 Infrared Command (for DIPX-5100/IPX6000)

Command	`infrared [-l N [-t T]] "IRDATA" hostname`
Response	`infrared [-l N [-t T]] "IRDATA" hostname`
Description	Transmits infrared information "IRDATA" to the target device; the infrared information should be within the double quotations (""). The option -l is used to designate the number of the times the unit emits the IR code repeatedly, its argument N is the times, the legal range is from 1 to 10. When this option is absent, the unit emits the IR code only once. When the option -l exists, an additional option -t can be used to designated the interval between the two consecutive emitting. Its unit is millisecond. When this option is absent, the default is 20ms. So far, the above options are available for IPX6000 only. **Note:** • This command is available for DIPX-5100 • DIPX-5100 supports "IRDATA" in both pronto and global cache formats. • IPX6000 supports "IRDATA" in both pronto and global cache formats.

Example 1:

If you want to transmit IR information to DIPX-5100:

Command (pronto format):

```
infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018 0018
0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018
0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE"
OUT2-MONITOR
```

Response:

```
infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018 0018
0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018
0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE"
OUT2-MONITOR
```

Example 2:

If you want to transmit IR information to DIPX-5100:

Command (global cache format):

```
infrared
"40076,1,1,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,6
7,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19,
67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,2
4,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,
67,19,24,19,67,19,67,19,24,19,24,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19
,67,19,1811,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,
67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19
,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,
24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19
,67,19,24,18,1" OUT2-MONITOR
```

Response:

```
infrared
"40076,1,1,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,6
7,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19,
67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,2
4,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,
67,19,24,19,67,19,67,19,24,19,24,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19
,67,19,1811,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,
67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19
,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,
24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19
,67,19,24,18,1" OUT2-MONITOR
```

Note: After you send this command to the IP controller, the IP controller converts it into the *irs* command and then forward this *irs* command to the target device.

Example 3:

If you want to instruct an IPX6000 to emit IR signal 3 times with the interval of 50ms:

Command (pronto format):

```
infrared -l 3 -t 50 "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018
0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018
0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE"
OUT2-MONITOR
```

Response:

```
infrared -l 3 -t 50 "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018
0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018
0018 0018 0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE"
OUT2-MONITOR
```

Command	infrared "*IRDATA*" *hostname*
Response	infrared "*IRDATA*" *hostname*
Description	Transmits infrared information “IRDATA” to the target device; the infrared information should be within the double quotations (""). **Note:**

	• This command is available for DIPX-5100 and IPX6000 only. • DIPX-5100 supports “IRDATA” in both pronto and global cache formats. • IPX6000 supports “IRDATA” in pronto format only.

Example 1:

If you want to transmit IR information to DIPX-5100:

Command (pronto format):

```
infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018 0018 0030
0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018 0018 0018
0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE" OUT2-MONITOR
```

Response:

```
infrared "0000 0067 0000 0015 0060 0018 0030 0018 0018 0018 0030 0018 0018 0018 0030
0018 0018 0018 0018 0018 0018 0018 0030 0018 0018 0018 0030 0018 0030 0018 0018 0018
0030 0018 0018 0018 0018 0018 0018 0018 0030 0018 0030 0018 0030 01FE" OUT2-MONITOR
```

Example 2:

If you want to transmit IR information to DIPX-5100:

Command (global cache format):

```
infrared "40076,1,1,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,
19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24
,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,
19,67,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24
,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19,67,19,24,19,67,19,24,19,24,19,6
7,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67
,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,2
4,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67
,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,2
4,19,24,19,67,19,24,18,1" OUT2-MONITOR
```

Response:

```
infrared "40076,1,1,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67,
19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,24
,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67,
19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,24
,19,24,19,67,19,24,19,67,19,67,19,24,19,24,19,24,19,67,19,24,19,67,19,24,19,24,19,6
7,19,67,19,67,19,1811,170,174,19,67,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,67
,19,67,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,24,19,67,19,67,19,24,19,2
4,19,24,19,67,19,24,19,67,19,24,19,24,19,67,19,67,19,67,19,1811,170,174,19,67,19,67
,19,67,19,24,19,24,19,24,19,24,19,24,19,24,19,67,19,67,19,67,19,24,19,24,19,24,19,2
4,19,24,19,67,19,24,18,1" OUT2-MONITOR
```

Note: After you send this command to the IP controller, the IP controller converts it into the *irs* command and then forward this *irs* command to the target device.

2.6 cec Commands

Command	cec "*CECDATA*" *hostname*
Response	cec "*CECDATA*" *hostname*

Description	Transmits CEC information “CECDATA” to the device's hostname, the CEC information should be within the double quotation marks ("").

Example:

If you want to transmit cec information to the device:

Command:

```
cec "ff36" OUT2-MONITOR
```

Response:

```
cec "ff36" OUT2-MONITOR
```

2.7 Update Commands (for IPX6000)

Command	update*firmwarename dev1...*
Response	update*firmwarename dev1... {error|processing} ["errorinfo"]*
	Update the device’s firmware. When the *firmware name* doesn’t exist, an error will return with error information. **Note:** 1. To update the device’s firmware through the IP controller, perform the following: a) Upload the device’s firmware to the specified directory of the IP controller using the URL *http://ip/update_device* b) Send the ‘update’ command to the IP controller. The IP controller transfers the firmware to the corresponding device(s) for updating. 2. According to the IP6000 hardware architecture, this command can only update the FPGA microcode, i.e. it cannot update the firmware of any external component.

If you want to update device’s firmware:

Example 1:

Command:

```
update firmware_V1.2.1.tar.gz RX1 RX2 RX3
```

Response:

```
update firmware_V1.2.1.tar.gz RX1 RX2 RX3 processing
```

Example 2:

Command:

```
update firmware_V1.2.1.tar.gz RX1 RX2 RX3
```

Response:

```
update firmware_V1.2.1.tar.gz RX1 RX2 RX3 error "file not exist"
```

2.8 canvas Commands (for DIPD-5100)

2.8.1 canvas get

Command	`canvas get [canvas1]`

Response	`canvas layout list:\r\n`
	`canvas1 layout11 layout12…layout1m\r\n`
	`canvas2 layout21 layout22…layout2m\r\n`
	…
	`canvasn layoutn1 layoutn2…layoutnm\r\n`
	`\r\n`

Description	Get the layout name of the mosaic style video walls.

Example:

Command:

```
canvas get
```

Response:

```
canvas layout list:\r\n
canvas1 layout1\r\n
canvas2 layout1\r\n
canvas3 layout1\r\n
\r\n
```

2.8.2 canvas getjson

Command	`canvas getjson [canvas1]`
Response	`canvas json string:\r\n` *jsonstring* *\r\n*
Description	Get the json information of the specified mosaic style video wall.

If you want to get the json information of canvas1:

Example:

Command:

```
canvas getjson canvas1
```

Response:

```
canvas json string:
[
    {
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 1
```

```
    }
],
"hsize" : 640,
"layouts" : [
    {
        "layoutseq" : 1,
        "maxwindownum" : 16,
        "name" : "Layout1",
        "windows" : [
            {
                "hsize" : 640,
                "hstart" : 0,
                "layerseq" : 1,
                "name" : "1",
                "stretchtype" : "stretch",
                "tx" : "DIPE5100-341B228084C2",
                "vsize" : 360,
                "vstart" : 0
            }
        ]
    }
],
"name" : "canvas1",
"rxArray" : [
    {
        "aliasName" : "IPD5100-341B22800D20",
        "brh" : 395,
        "brv" : 203,
        "marginbotton" : 0,
        "marginleft" : 14,
        "marginright" : 14,
        "margintop" : 10,
        "rotate" : 0,
        "tlh" : 215,
        "tlv" : 88,
        "trueName" : "IPD5100-341B22800D20"
    }
```

```
            ],
            "sequence" : 5,
            "vsize" : 360
        }
    ]
```

2.8.3 canvas layout getjson

Command	`canvas layout getjson` *`canvasname layoutname`*
Response	`canvas layout json string:\r\n` *`jsonstring`* *`\r\n`*
Description	Get a layout's json information for the specified canvas

If you want to get the json information of Layout 1 for the specified mosaic style video wall:

Example:

Command:

```
canvas getjson canvas1 Layout1
```

Response:

```
canvas layout json string:
{
    "layoutseq" : 1,
    "maxwindownum" : 16,
    "name" : "Layout1",
    "windows" : [
        {
            "hsize" : 640,
            "hstart" : 0,
            "layerseq" : 1,
            "name" : "1",
            "stretchtype" : "stretch",
            "tx" : "DIPE5100-341B228084C2",
            "vsize" : 360,
            "vstart" : 0
        }
    ]
}
```

2.8.4 canvas active

Command	`canvas active` *`canvasname layoutname`*

Response	canvas active *canvasname layoutname* {success|failure}
Description	Apply one specified layout to the corresponding mosaic style video wall.

Example:

Command:

```
canvas active canvas1 Layout1
```

Response:

```
canvas active canvas1 Layout1 success
```

2.8.5 canvas changetx

Command	canvas changetx *canvasname layoutname txname*
Response	canvas changetx *canvasname layoutname txname* {success|failure}
Description	Assign a video source to all windows of the specified layout in the corresponding mosaic video wall, and apply this layout.

Example:

Command:

```
canvas changetx canvas1 Layout1 DIPE5100-341B2280C570
```

Response:

```
canvas changetx canvas1 Layout1 DIPE5100-341B2280C570 success
```

2.8.6 canvas window changetx

Command	canvas window changetx *canvasname layoutname windowname txname*
Response	canvas window changetx *canvasname layoutname windowname txname* {success|failure}
Description	Assign a video source to the specified window of the specified layout in the mosaic video wall.

Example:

Command:

```
canvas window changetx canvas1 Layout1 window1 DIPE5100-341B2280C570
```

Response:

```
canvas window changetx canvas1 Layout1 window1 DIPE5100-341B2280C570 success
```

2.8.7 canvas group update

Command	canvas group update *[json]*
Response	canvas group or sort update success

Description	Arrange the group and the mosaic style video wall in sequence. Note: Since the default sequence of the group “ungrouped” is 1, the sequence of the newly created group must start from “2”.

Example:

Command:

```
canvas group update
[{"group":[{"name":"group1","sequence":2}],"sequence":3,"name":"canvas"}]
```

Response:

```
canvas group or sort update success
```

2.8.8 canvas screenproperties getjson

Command	`canvas screenproperties getjson` *`brand model`*
Response	`canvas screenproperties json string:\r\n` *`jsonstring`* *`\r\n`*
Description	Obtain the attribute information in json format of the TVs that make up the mosaic style video wall. The *brand model* is optional which denotes the TV’s brand, if the brand specified, the controller will respond with the TV brand’s attribute information.

Example:

Command:

```
canvas screenproperties getjson
```

Response:

```
canvas screenproperties json string:
[
    {
        "brand" : "ddd",
        "height" : 720,
        "marginbotton" : 10,
        "marginleft" : 10,
        "marginright" : 10,
        "margintop" : 10,
        "model" : "fff",
        "size" : 65,
        "width" : 1280
    }
]
```

2.8.9 canvas screenproperties create

Command	canvas screenproperties create *canvasjsonstring*
Response	canvas Screenproperties create success
Description	Create or update the attribute information in json format of the TVs that make up the mosaic style video wall.

Example:

Command:

```
canvas Screenproperties create
[
    {
        "brand" : "ddd",
        "height" : 720,
        "marginbotton" : 10,
        "marginleft" : 10,
        "marginright" : 10,
        "margintop" : 10,
        "model" : "fff",
        "sequence" : 0,
        "size" : 65,
        "width" : 1280
    }
]
```

Response:

```
canvas Screenproperties create success
```

2.8.10 canvas screenproperties remove

Command	canvas screenproperties remove *brand1 model1*
Response	canvas Screenproperties remove *brand1 model1* success
Description	Remove the TV screen, including brand and model no. The brand and model no. shall be separated with space. Only one TV can be removed each time.

Example:

Command:

```
canvas screenproperties remove brand1 model1
```

Response:

```
canvas screenproperties remove brand1 model1 success
```

2.8.11 canvas create

Command	canvas create *[canvasjsonstring]*
Response	canvas create success
Description	Update the layout that exists in controller using the json information *[canvasjsonstring].*

Example:

Command:

```
canvas create
canvas json string:
[
    {
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 1
            }
        ],
        "hsize" : 10240,
        "layouts" : [
            {
                "layoutseq" : 1,
                "maxwindownum" : 16,
                "name" : "Layout1",
                "windows" : [
                    {
                        "hsize" : 10240,
                        "hstart" : 0,
                        "layerseq" : 1,
                        "name" : "1",
                        "stretchtype" : "stretch",
                        "tx" : "NULL",
                        "vsize" : 5760,
                        "vstart" : 0
                    }
                ]
            }
        ],
```

```
            "name" : "wwww",
            "rxArray" : [
                {
                    "aliasName" : "",
                    "brh" : 1615,
                    "brv" : 1045,
                    "marginbotton" : 10,
                    "marginleft" : 10,
                    "marginright" : 10,
                    "margintop" : 10,
                    "rotate" : 0,
                    "tlh" : 335,
                    "tlv" : 325,
                    "trueName" : ""
                },
                {
                    "aliasName" : "",
                    "brh" : 2320,
                    "brv" : 1815,
                    "marginbotton" : 10,
                    "marginleft" : 10,
                    "marginright" : 10,
                    "margintop" : 10,
                    "rotate" : 0,
                    "tlh" : 1040,
                    "tlv" : 1095,
                    "trueName" : ""
                }
            ],
            "sequence" : 3,
            "vsize" : 5760
        }
    ]
```


Response:

```
    canvas create success
```

2.8.12 canvas modify name

Command	canvas modify name *canvasname_old canvasname_new*
Response	canvas modify name *canvasname_old canvas name_new* success
Description	Modify the name of the mosaic style video wall.

Example:

If you want to modify the scene name of canvas1 to canvas2:

Command:

```
canvas modify name canvas1 canvas2
```

Response:

```
canvas modify name canvas1 canvas2 success
```

2.8.13 canvas remove

Command	canvas remove *canvasname1 canvasname2 ..*
Response	canvas remove *canvasname1 canvasname2 ..* success
Description	Remove the specified scene names.

Example:

If you want to modify the scene “canvas1”:

Command:

```
canvas remove canvas1
```

Response:

```
canvas remove canvas1 success
```

2.9 mrscene Commands (forDIPD-5100)

2.9.1 mrscene get

Command	mrscene get *[mrscene1]*
Response	mrscene list:\r\n mrscene1 layout11 layout12…layout1m\r\n mrscene2 layout21 layout22…layout2m\r\n … mrscenen layoutn1 layoutn2…layoutnm\r\n \r\n
Description	Obtains the layout names of the corresponding mouse roaming scenes.

Example:

If you want to obtain layout names of all mouse roaming scenes:

Command:

```
mrscene get
```

Response:

```
mrscene list:\r\n
mrscene1 layoutl\r\n
mrscene2 layout1\r\n
mrscene3 layout1\r\n
\r\n
```

2.9.2 mrscene getjson

Command	`mrscene getjson` *`[mrscene1]`*

Response	`mrscene json string:\r\n`
	`jsonstring`
	`\r\n`

Description	Obtains the information in json string of the specified mouse roaming scene.

Example:

If you want to obtain the json information of a specified mouse roaming scene:

Command:

```
mrscene getjson mrscene1
```

Response:

```
mrscene json string:
[
    {
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 1
            }
        ],
        "layoutseq" : 1,
        "m" : 2,
        "n" : 2,
        "name" : "weeee-3444",
        "rxArray" : [
            [
                {
                    "aliasName" : "IPD5100-2",
```

```
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
            "online" : true,
            "pos_x" : 0,
            "pos_y" : 0,
            "preview_enable" : "y",
            "sequence" : 1,
            "trueName" : "IPD5100-341B22802B41",
            "txName" : ""
        },
        null
    ],
    [
        null,
        {
            "aliasName" : "IPD5100-1",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
            "online" : true,
            "pos_x" : 1,
            "pos_y" : 1,
            "preview_enable" : "y",
            "sequence" : 2,
            "trueName" : "IPD5100-341B22802C09",
            "txName" : ""
```

```
                }
            ]
        ],
        "sceneAutoApply" : false,
        "sequence" : 2
    }
]
```

2.9.3 mrscene active

Command	mrscene active *mrscene1*
Response	mrscene active *mrscene1* {success|failure}
Description	Apply mouse roaming for the specified layout on the video wall.

Example:

If you want to apply mouse roaming for Layout 1:

Command:

```
mrscene active Layout1
```

Response:

```
mrscene active Layout1 success
```

2.9.4 mrscene create

Command	mrscene create *[mrscenejsonstring]*
Response	mrscene create success
Description	Create a mouse roaming scene that defined by *[mrscenejsonstring]* in server.

Example:

If you want to create a mouse roaming scene defined in json string for a device:

Command:

```
mrscene create
[
    {
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 1
            }
        ],
        "layoutseq" : 1,
```

```
"m" : 2,
"n" : 2,
"name" : "weeee-3444",
"rxArray" : [
    [
        {
            "aliasName" : "IPD5100-2",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
            "online" : true,
            "pos_x" : 0,
            "pos_y" : 0,
            "preview_enable" : "y",
            "sequence" : 1,
            "trueName" : "IPD5100-341B22802B41",
            "txName" : ""
        },
        null
    ],
    [
        null,
        {
            "aliasName" : "IPD5100-1",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
```

```
                    "online" : true,
                    "pos_x" : 1,
                    "pos_y" : 1,
                    "preview_enable" : "y",
                    "sequence" : 2,
                    "trueName" : "IPD5100-341B22802C09",
                    "txName" : ""
                }
            ]
        ],
        "sceneAutoApply" : false,
        "sequence" : 2
    }
]
```

Response:

```
mrscene create success
```

2.9.5 mrscene update

Command	mrscene update *[mrscenejsonstring]*
Response	mrscene update success
Description	Use the configuration information that defined by *[mrscenejsonstring]* to update a mouse roaming scene in server and apply it to the specified device.

Example:

If you want to update and apply a mouse roaming scene to a device:

Command:

```
mrscene update
[
    {
        "group" : [
            {
                "name" : "ungrouped",
                "sequence" : 1
            }
        ],
        "layoutseq" : 1,
        "m" : 2,
```

```
"n" : 2,
"name" : "weeee-3444",
"rxArray" : [
    [
        {
            "aliasName" : "IPD5100-2",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
            "online" : true,
            "pos_x" : 0,
            "pos_y" : 0,
            "preview_enable" : "y",
            "sequence" : 1,
            "trueName" : "IPD5100-341B22802B41",
            "txName" : ""
        },
        null
    ],
    [
        null,
        {
            "aliasName" : "IPD5100-1",
            "deviceType" : "Receiver",
            "group" : [
                {
                    "name" : "ungrouped",
                    "sequence" : 1
                }
            ],
            "mouse_roaming" : 0,
            "online" : true,
```

```
                    "pos_x" : 1,
                    "pos_y" : 1,
                    "preview_enable" : "y",
                    "sequence" : 2,
                    "trueName" : "IPD5100-341B22802C09",
                    "txName" : ""
                }
            ]
        ],
        "sceneAutoApply" : false,
        "sequence" : 2
    }
]
```


Response:

```
mrscene update success
```

2.9.6 mrscene modify name

Command	`mrscene modify name` *`scenename_old scenename_new`*
Response	`mrscene modify name` *`scenename_old scenename_new`* `success`
Description	Change the name of a mouse roaming scene to a new one. • *scenename_old*: The current name of the mouse roaming scene. • *scenename_new*: The new name of the mouse roaming scene.

Example:

If you want to change the name of the mouse roaming scene from *vw-scene1 to vw-scene2*:

Command:

```
mrscene modify name vw-scene1 vw-scene2
```

Response:

```
mrscene modify name vw-scene1 vw-scene2 success
```

2.9.7 mrscene remove

Command	`mrscene remove` *`scenename1 scenename2 ..`*
Response	`mrscene remove` *`scenename1 scenename2 ..`* `success`
Description	Remove the specified mouse roaming scene(s).

Example:

If you want to remove the mouse roaming scene *vw-scene1*:

Command:

```
mrscene remove vw-scene1
```

Response:

```
mrscene remove vw-scene1 success
```

2.9.8 mrscene group update

Command	`mrscene group update` *`[json]`*
Response	`mrscene group or sort update success`
Description	Update the group information of mouse roaming scenes.

Example:

If you want to update the group information of the mouse roaming scene:

Command:

```
mrscene group update
[{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"name":"scene2-2x22"}]
```

Response:

```
mrscene group or sort update success
```

3. Concurrent Commands

A concurrent command, separated by semicolons (;), contains two or more commands. Itis designed to implement multiple operations at a time using one command, which is commonly used in matrix switching, video wall switching and scene switching. The following table lists the commands that can be used as concurrent commands.

Commands
matrix set TX1 RX1 RX2, TX2 RX3 RX4, ...
vw change rx tx
vw change vw-name tx
scene active scenename
scene connect scenename tx1 tx2 … txnm

Example:

If you want to implement three commands at the same time using a concurrent command:

Command:

```
matrix set tx1 rx11 rx12 rx13; vw change vwtest2 DIPE5100-341B22430115; scene
connect scene1 tx1 tx2 tx3 tx4
```

3.1 Device Info

config set device info and **config get device info** send data in **key-value** format, **key** is parameter and **value** is its value. The following table lists the parameters supported by devices and their value ranges. All parameters can be changed,

unless otherwise stated.

Parameters	Description	Devices Supported
name	Device name. Read only. Format is "Device type-MAC address" such as DIPE5100-341B22FFFFB3	All devices
version	Device software version. Read only. Format is v#.#.# such as v2.5.6	All devices
mac	Device MAC address. Read only.	All devices
ip_mode	IP address obtain method. • 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
sourcein	Input port. For more information, see source commands.	Tx
enc_rc_mode	Data rate control method. cbr is CBR mode. vbr is VBR mode. fixqp is Fixed QP mode.	Tx
profile	Encoding profile. bp is base profile. mp is main profile. hp is high profile.	Tx
cbr_avg_bitrate	CBR coding average rate. Unit is kbps. Data rate of DIPE5100 is less than or equal to 30720.	Tx
vbr_max_bitrate	VBR encoding maximum rate. Unit is kbps. Data rate of DIPE5100 is less than or equal to 30720.	Tx
vbr_min_gp	VBR minimum quantification parameters. Range is [0, 51].	Tx
vbr_max_gp	VBR maximum quantification parameters. Range is [0, 51].	Tx
fixqp_iqp	FixQP encoding mode I-frame quantification parameters. Range is [0, 51].	Tx
fixqp_pqp	FixQP encoding mode P-frame quantification parameters. Range is [0, 51].	Tx
enc_gop	GOP size. Range is [1, 65535]. There is one I-frame in a specific range.	Tx
enc_fps	Frames per second. Range is [1, 60].	Tx
transport_type	Streaming media encapsulation format. raw is private format. ts is MPEG-2 TS format.	Tx
audio.name	Audio interface name. Read-only. 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.	Rx

3.1.1 IPD5100/IPD5100L

Parameter	Description	Example
output_force_resolution	Configure video output resolution: • pass-through • fixed resolution: 2160p30, 2160p25, 2160p24, 1080p60, 1080p50, 1080p30, 1080p25, 1080p24, 720p60, 720p50, 720p30, 720p25, 720p24 • Auto	output_force_resolution=1080p60
hdcp_hybrid_mode	original | always | always22	hdcp_hybrid_mode= always
turn_off_screen	Determine whether to turn off screen when there's no source. y/n	turn_off_screen=y

Parameter	Description	Example
src_unavailable_timeout	The timeout that detects signal lost. Range: [1,60] -1: Timeout doesn't occur	src_unavailable_timeout=50
video_stretch_type	stretch_out/fit_in	video_stretch_type=fit_in
video_rotate	Video rotation angle: 0°/180°/270°	video_rotate=180
video_gen_lock	Video genlock setting, [0-FFFFFFFF]	video_gen_lock=22334455
audio_gen_lock	Audio genlock setting, [0-FFFFFFFF]	audio_gen_lock=22334455
enc_fps	[0,60]	enc_fps=60
rs232_baudrate	115200-8n1	rs232_baudrate=115200-8n1
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through
km_over_ip_enable	true/false	km_over_ip_enable=true
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true
audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable=true
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false
sinkpower.cec_mode	To enable CEC control on sinkpower: y/n	sinkpower.cec_mode=y
sinkpower.cec.onetouchplay	The CEC command value of OneTouchPlay; if there're multiple CEC commands, use comma " ," to connect them.	sinkpower.cec.onetouchplay="4004"
sinkpower.cec.standby	The CEC command value of Standby; if there're multiple CEC commands, use comma " ," to connect them.	sinkpower.cec.standby="ff36"
sinkpower.rs232_mode	To enable RS232 control on sinkpower: y/n	sinkpower.rs232_mode=y
sinkpower.rs232.mode	The format of the RS232 commands: hex/ascii	sinkpower.rs232.mode=ascii
sinkpower.rs232.param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	sinkpower.rs232.param=9600-8n1
sinkpower.rs232.onetouchplay	RS232 command that used to control OneTouchPlay.	sinkpower.rs232.onetouchplay="Power on"
sinkpower.rs232.standby	RS232 command that used to control StandBy.	sinkpower.rs232.standby="Power off"
sinkpower.ir_mode	To enable IR control on sinkpower: y/n	sinkpower.ir_mode=y
sinkpower.ir.onetouchplay	IR command that is used to control OneTouchPlay. This command must be quoted by quotation mark ("") and be sent individually.	sinkpower.ir.onetouchplay="Power on"
sinkpower.ir.standby	IR command that is used to control Standby. This command must be quoted by quotation mark ("") and be	sinkpower.ir.standby="Power off"

Parameter	Description	Example
	sent individually.	
ldap.enable	To enable or disable LDAP settings. value: {y|n} • y: enable • n: disable	ldap.enable=y
ldap.mode	LDAP mode. value: {dn|uid}	ldap.mode=uid
ldap.url	Format: ldap[s]://hostname:port. Example 1: ldap://192.168.17.225:389 (non-encrypted) Example 2: ldaps://192.168.17.225:636 (SSL encrypted)	ldap.url=ldap://
ldap.uid	User ID	ldap.uid= jeremy
ldap.base_dn	Domain name, which divides the whole domain into several parts. e.g. dc=my-domain, dc=com	ldap.base_dn= dc=my-domain,dc=com
ldap.bind_dn	The unique identifier of distinguished name, which includes two types: cn-based, cn+ou+dc; uid-based, uid+ou+dc. e.g. cn=jeremy,ou=People,dc=my-domain,dc=com	ldap. bind_dn = cn=jeremy, ou=People,dc=my-domain,dc=com
ldap.user_attr	Attribute query, e.g. loginShell=/bin/bash, query whether loginShell is /bin/bash	ldap.user_attr=y
ldap.password	user password	ldap.password=123456
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} • y: enable • n: disable	ieee8021x.enable=y
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls
ieee8021x. mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user
ieee8021x. mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd
ieee8021x.tlsuser	TlS user name	ieee8021x.tlsuser=user
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikeypwd
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y
arp_input	ARC audio input: Toslink (default)/hdmiout. • Toslink: Toslink at RX is selected as the ARC audio input. • hdmiout: HDMI Out at RX is selected as the ARC audio input.	arp_input=toslink
arp_hdmiout_cap	Capability of the ARC audio – ARC/eARC. • ARC: Enable ARC for ARC audio. • eARC: Enable ARC and eARC.	arp_hdmiout_cap=arc
ui_show_text	To turn on/off OSD display, y/n.	ui_show_text=y

3.1.2 DIPE5100/DIPE5100DNT/DIPE5100W/DIPE5100L/DIPE5100P/DIPE5102/DIPE5101

Parameter	Description	Example
audio_input_type	Available audio input channels: auto, hdmi, analog	Read only.
audio_analog_out_vol	Range of the analog audio output volume: [0,100]	audio_analog_out_vol=90
audio_direction	Settings of audio direction: in, out. Available devices: DIPE5101/DIPE5102	audio_direction=in
dante_port	Assign Dante to specific Ethernet port: • eth0 • eth1 (Default value) Available devices: DIPE5101/DIPE5102 (Restart to take effect)	dante_port=eth1

Parameter	Description	Example
	On web UI, it corresponds to the configuration item “Using LAN2 as Independent Dante Port” on/off. • On: dante_port=eth1, Dante feature is turned on. • Off: dante_port=eth0, Dante feature is turned off. default setting is off.	
dante_ip_mode	Dante IP mode: Static, DHCP (default value) Available devices: DIPE5101/DIPE5102 (Restart to take effect)	Read only.
dante_ipaddr	Dante IP address Available devices: DIPE5101/DIPE5102 (Restart to take effect)	Read only.
dante_netmask	Dante IP netmask Available devices: DIPE5100P/DIPE5102 (Restart to take effect)	Read only.
preview_enable	To enable preview: y/n	preview_enable=y
video_quality	Settings of video quality: • Auto: -1 • Quality 0(The best): 0 • Quality 1: 1 • Quality 2: 2 • Quality 3: 3 • Quality 4: 4 • Quality 5 (The Worst): 5	video_quality=1
bit_rate	Settings of bit rate: auto/10M/50M/100M/150M/200M	Bit_rate=100M
hdcp_enable	• y: HDCP-encryption is enabled. • n: HDCP-encryption is disabled.	hdcp_enable=y
audio_addon	• none • dante	audio_addon=dante
dante_op_mode	Dante AV-A operation mode: • AV-A (default value) • dante Note: Restart to take effect.	dante_op_mode=AV-A
dante_trial_mode	To enable/disable Dante trial mode without Dante authentication (one hour for each time): • y: Dante trial mode is enabled. • n: Dante trial mode is disabled.	dante_trial_mode=y
audio_bridge_en	To enable/disable bridge between two audio networks. • y • n	audio_bridge_en=y
video_source_switch	Video source switching: HDMI/USB-C/none (3.5mm audio mode) Note: This parameter is available for DIPE5100W only and will take effect immediately on the setting change. If DIPE5100W reboots, it will be restored to HDMI (default).	video_source_switch=hdmi
usb_c_charger	USB-C charging function enabled or disabled. y: USB-C charging enabled. n: USB-C charging disabled. Note: This parameter is available for DIPE5100W only and will take effect immediately on the setting change. If DIPE5100W reboots, it will load the new setting. Default setting is n.	usb_c_charger=y
rs232_baudrate	Serial parameter that includes baud rate, data bits and parity bits. • Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. • Value of the data bits: 5, 6, 7, 8. • Value of the parity bits: o, e, n. • Value of the stop bits: 1, 2.	rs232_baudrate=9600-8n1
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through
rs232_token_timeout	This parameter is available when RS232 working mode is set as “pass through” and is disabled when RS232 working mode is set as “Feedback”. Available range: [0,10]	rs232_token_timeout=1
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true

audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable =true
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true
km_over_ip_enable	true/false	km_over_ip_enable=false

Parameter	Description	Example
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false
edid	Reset EDID.	edid=reset
edid	EDID Content. The EDID content shall be quoted by quotation marks and separated by spaces, and must be sent individually.	edid="00 ff ff ff ff ff ff 00 06 74 01 00 01 00 00 00 0a 1c 01 03 80 00 00 78 0a ee 95 a3 54 4c 99 26 0f 50 54 a1 08 00 31 40 45 40 61 40 71 40 81 80 01 00 01 00 01 01 08 e8 00 30 f2 70 5a 80 b0 58 8a 00 50 1d 74 00 00 1e 02 3a 80 18 71 38 2d 40 58 2c 45 00 50 1d 74 00 00 1e 00 00 00 fd 00 0f 90 1e 88 3c 00 00 00 00 00 00 00 00 00 00 00 fc 00 56 45 52 54 45 58 0a 20 20 20 20 20 20 01 6e 02 03 5b 71 57 61 10 1f 60 13 05 14 20 21 22 5d 5e 5f 04 65 66 62 63 64 07 16 03 12 35 0f 7f 07 15 07 50 3d 1e c0 4d 02 00 57 06 01 5f 7e 01 67 7e 01 83 5f 00 00 6e 03 0c 00 10 00 b8 3c 20 80 80 01 02 03 04 67 d8 5d c4 01 78 80 00 e2 00 cf e2 0f 09 e3 06 0f 01 e3 05 e3 01 01 1d 80 18 71 1c 16 20 58 2c 25 00 40 84 63 00 00 9e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16"
ldap.enable	To enable or disable LDAP settings. value: {y|n} • y: enable • n: disable	ldap.enable=y
ldap.mode	LDAP mode. value: {dn|uid}	ldap.mode=uid
ldap.url	Format: ldap[s]://hostname:port. Example 1: ldap://192.168.17.225:389 (non-encrypted) Example 2: ldaps://192.168.17.225:636 (SSL encrypted)	ldap.url=ldap://
ldap.uid	User ID	ldap.uid= jeremy
ldap.base_dn	Domain name, which divides the whole domain into several parts. e.g. dc=my-domain, dc=com	ldap.base_dn= dc=my-domain,dc=com
ldap.bind_dn	The unique identifier of distinguished name, which includes two types: cn-based, cn+ou+dc; uid-based, uid+ou+dc. e.g. cn=jeremy,ou=People,dc=my-domain,dc=com	ldap. bind_dn = cn=jeremy, ou=People,dc=my-domain,dc=com
ldap.user_attr	Attribute query, e.g. loginShell=/bin/bash, query whether loginShell is /bin/bash	ldap.user_attr=y
ldap.password	user password	ldap.password=123456
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} • y: enable • n: disable	ieee8021x.enable=y
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls
ieee8021x. mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user
ieee8021x. mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd
ieee8021x.tlsuser	TlS user name	ieee8021x.tlsuser=user
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikeypwd
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y
dante.audio_input	Dante audio source input – hdmi/analog Note: This parameter is available for DIPE5100DNT only.	dante.audio_input=analog
dante.net_mode	Dante network mode – normal/isolate Note: This parameter is available for DIPE5100DNT only.	dante.net_mode=normal

3.1.3 IPD5100HYB

Parameter	Description	Example
output_force_resolution	Configure video output resolution: • pass-through • fixed resolution: 2160p30, 2160p25, 2160p24, 1080p60, 1080p50, 1080p30, 1080p25, 1080p24, 720p60, 720p50, 720p30, 720p25, 720p24 • Auto	output_force_resolution=1080p60
hdcp_hybrid_mode	original | always | always22	hdcp_hybrid_mode= always
turn_off_screen	Determine whether to turn off screen when there's no source. y/n	turn_off_screen=y
src_unavailable_timeout	The timeout that detects signal lost. Range: [1,60] -1: Timeout doesn't occur	src_unavailable_timeout=50
video_stretch_type	stretch_out/fit_in	video_stretch_type=fit_in
video_rotate	Video rotation angle: 0°/180°/270°	video_rotate=180
video_gen_lock	Video genlock setting, [0-FFFFFFFF]	video_gen_lock=22334455
audio_gen_lock	Audio genlock setting, [0-FFFFFFFF]	audio_gen_lock=22334455
enc_fps	[0,60]	enc_fps=60
rs232_baudrate	115200-8n1	rs232_baudrate=115200-8n1
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true
audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable=true
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true
km_over_ip_enable	true/false	km_over_ip_enable=true
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false
sinkpower.rs232_mode	To enable RS232 control on sinkpower: y/n	sinkpower.rs232_mode=y
sinkpower.rs232.mode	The format of the RS232 commands: hex/ascii	sinkpower.rs232.mode=ascii
sinkpower.rs232.param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	sinkpower.rs232.param=9600-8n1
sinkpower.rs232.onetouchplay	RS232 command that used to control OneTouchPlay.	sinkpower.rs232.onetouchplay="Power on"
sinkpower.rs232.standby	RS232 command that used to control StandBy.	sinkpower.rs232.standby="

Parameter	Description	Example
		Power off"
sinkpower.ir_mode	To enable IR control on sinkpower: y/n	sinkpower.ir_mode=y
sinkpower.ir.onetouchplay	IR command that is used to control OneTouchPlay. This command must be quoted by quotation mark (“”) and be sent individually.	sinkpower.ir.onetouchplay= "Power on"
sinkpower.ir.standby	IR command that is used to control Standby. This command must be quoted by quotation mark (“”) and be sent individually.	sinkpower.ir.standby="Pow er off"
ldap.enable	To enable or disable LDAP settings. value: {y|n} • y: enable • n: disable	ldap.enable=y
ldap.mode	LDAP mode. value: {dn|uid}	ldap.mode=uid
ldap.url	Format: ldap[s]://hostname:port. Example 1: ldap://192.168.17.225:389 (non-encrypted) Example 2: ldaps://192.168.17.225:636 (SSL encrypted)	ldap.url=ldap://
ldap.uid	User ID	ldap.uid= jeremy
ldap.base_dn	Domain name, which divides the whole domain into several parts. e.g. dc=my-domain, dc=com	ldap.base_dn= dc=my-domain,dc=com
ldap.bind_dn	The unique identifier of distinguished name, which includes two types: cn-based, cn+ou+dc; uid-based, uid+ou+dc. e.g. cn=jeremy,ou=People,dc=my-domain,dc=com	ldap. bind_dn = cn=jeremy, ou=People,dc=my-domain,dc=com
ldap.user_attr	Attribute query, e.g. loginShell=/bin/bash, query whether loginShell is /bin/bash	ldap.user_attr=y
ldap.password	user password	ldap.password=123456
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} • y: enable • n: disable	ieee8021x.enable=y
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls
ieee8021x. mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user
ieee8021x. mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd
ieee8021x.tlsuser	TlS user name	ieee8021x.tlsuser=user
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikeypwd
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y
ui_show_text	To turn on/off OSD display, y/n.	ui_show_text=y

3.1.4 DIPE5100HYB

Parameter	Description	Example
audio_input_type	Available audio input channels: hdmi (default), analog	Read only.
preview_enable	To enable preview: y/n	preview_enable=y
video_quality	Settings of video quality: • Auto: -1 • Quality 0(The best): 0 • Quality 1: 1	video_quality=1

Parameter	Description	Example
	• Quality 2: 2 • Quality 3: 3 • Quality 4: 4 • Quality 5 (The Worst): 5	
bit_rate	Settings of bit rate: auto/10M/50M/100M/150M/200M	Bit_rate=100M
hdcp_enable	• y: HDCP-encryption is enabled. • n: HDCP-encryption is disabled.	hdcp_enable=y
rs232_baudrate	Serial parameter that includes baud rate, data bits and parity bits. • Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. • Value of the data bits: 5, 6, 7, 8. • Value of the parity bits: o, e, n. • Value of the stop bits: 1, 2.	rs232_baudrate=9600-8n1
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through
rs232_token_timeout	This parameter is available when RS232 working mode is set as “pass through” and is disabled when RS232 working mode is set as “Feedback”. Available range: [0,10]	rs232_token_timeout=1
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true
audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable =true
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true
km_over_ip_enable	true/false	km_over_ip_enable=false
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false
edid	Reset EDID	edid=reset
edid	EDID Content. The EDID content shall be quoted by quotation marks and separated by spaces, and must be sent individually.	edid=“00 ff ff ff ff ff ff 00 06 74 01 00 01 00 00 00 0a 1c 01 03 80 00 00 78 0a ee 95 a3 54 4c 99 26 0f 50 54 a1 08 00 31 40 45 40 61 40 71 40 81 80 01 00 01 00 01 01 08 e8 00 30 f2 70 5a 80 b0 58 8a 00 50 1d 74 00 00 1e 02 3a 80 18 71 38 2d 40 58 2c 45 00 50 1d 74 00 00 1e 00 00 00 fd 00 0f 90 1e 88 3c 00 00 00 00 00 00 00 00 00 00 00 fc 00 56 45 52 54 45 58 0a 20 20 20 20 20 20 01 6e 02 03 5b 71 57 61 10 1f 60 13 05 14 20 21 22 5d 5e 5f 04 65 66 62 63 64 07 16 03 12 35 0f 7f 07 15 07 50 3d 1e c0 4d 02 00 57 06 01 5f 7e 01 67 7e 01 83 5f 00 00 6e 03 0c 00 10 00 b8 3c 20 80 80 01 02 03 04 67 d8 5d c4 01 78 80 00 e2 00 cf e2 0f 09 e3 06 0f 01 e3 05 e3 01 01 1d 80 18 71 1c 16 20 58 2c 25 00 40 84 63 00 00 9e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16”
ldap.enable	To enable or disable LDAP settings. value: {y|n} • y: enable • n: disable	ldap.enable=y
ldap.mode	LDAP mode. value: {dn|uid}	ldap.mode=uid
ldap.url	Format: ldap[s]://hostname:port. Example 1: ldap://192.168.17.225:389 (non-encrypted) Example 2: ldaps://192.168.17.225:636 (SSL encrypted)	ldap.url=ldap://
ldap.uid	User ID	ldap.uid= jeremy
ldap.base_dn	Domain name, which divides the whole domain into several parts.	ldap.base_dn= dc=my-domain,dc=com

Parameter	Description	Example
	e.g. dc=my-domain, dc=com	
ldap.bind_dn	The unique identifier of distinguished name, which includes two types: cn-based, cn+ou+dc; uid-based, uid+ou+dc. e.g. cn=jeremy,ou=People,dc=my-domain,dc=com	ldap. bind_dn = cn=jeremy, ou=People,dc=my-domain,dc=com
ldap.user_attr	Attribute query, e.g. loginShell=/bin/bash, query whether loginShell is /bin/bash	ldap.user_attr=y
ldap.password	user password	ldap.password=123456
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} • y: enable • n: disable	ieee8021x.enable=y
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls
ieee8021x. mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user
ieee8021x. mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd
ieee8021x.tlsuser	TlS user name	ieee8021x.tlsuser=user
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikey pwd
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y

3.1.5 IPD6000V3

Parameters for IPD6000V3 only:

Parameter	Description	Example
video_mode	The video output mode includes genlock， fast_switch and genlock_scaling.	video_mode=genlock
video_timing	Video output resolution, which is available only if the video_mode is set as fast_switch. The value of the parameter includes the following: 720P@50,720P@60, 1080P@50, 1080P@60,2160P@25,2160P@30,2160P@50,2160P@60, 4096x2160@24, 4096x2160@30, 4096x2160@60, 1024x768@60, 1280x768@60, 1280x960@60, 1280x1024@60, 1360x768@60, 1400x1050@60, 1600x1200@60, 1680x1050@60, 1920x1200@60	video_timing=720P@50
genlock_scaling_resolution	Resolution setting in genlock_scaling mode: Auto, 720P, 1080P, 2160P, 4096x2160, 1024x768, 1280x768, 1280x960, 1280x1024, 1360x768, 1400x1050, 1600x1200, 1680x1050, 1920x1200	genlock_scaling_resolution=AUTO
video_stretch_type	Video stretch mode, which is available only if the video_mode is set as fast_switch. The value of the parameter includes none, fit_in and stretch_out.	video_stretch_type=fit_in
hdmi_audio_source	HDMI audio source. The value of the parameter includes hdmi, analog and dmix.	hdmi_audio_source=hdmi
analog_audio_source	Analog audio source. The value of the parameter includes analog and dmix.	analog_audio_source=dmix
serial_param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200.	serial_param=9600-8n1

Parameter	Description	Example
	Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	
edid	EDID content	Read only
sinkpower.mode	Three modes: All, CEC, RS232.	sinkpower.mode=ALL
sinkpower.cec.onetouchplay	The value of “OneTouchPlay” controlled by CEC, if there’re combined CEC commands, each two different commands would be connected by a commas (,).	sinkpower.cec.onetouchplay=4004
sinkpower.cec.standby	The value of “StandBy” controlled by CEC, if there’re combined CEC commands, each two different commands would be connected by a commas (,).	sinkpower.cec.standby=ff36
sinkpower.rs232.cmdmode	RS232 control mode, includes hex and ascii.	sinkpower.rs232.cmdmode=ascii
sinkpower.rs232.param	RS232 serial parameter. The parameter includes baud rate, data bits, parity bits and stop bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	sinkpower.rs232.param=9600-8n1
sinkpower.rs232.onetouchplay	RS232 command that used to control OneTouchPlay.	sinkpower.rs232.onetouchplay= "Power on"
sinkpower.rs232.standby	RS232 command that used to control StandBy.	sinkpower.rs232.standby="Power off"
icronusb.ip.mode	static/hdcp	icronusb.ip.mode=dhcp
icronusb.ip.addr	like 192.168.11.243	icronusb.ip.addr=192.168.11.243
icronusb.ip.netmask	like 255.255.0.0	netmask=255.255.0.0
icronusb.ip.gateway	like 192.168.11.1	gateway=192.168.11.1

3.1.6 IPD6000L(F)

Parameters for IPD6000L&IPD6000LF only:

Parameter	Description	Example
video_mode	The video output mode: genlock.	Read only
video_stretch_type	Video stretch mode: none.	Read only
hdmi_audio_source	HDMI audio source. The value of the parameter includes hdmi, analog and dmix.	hdmi_audio_source=hdmi
analog_audio_source	Analog audio source. The value of the parameter includes analog and dmix.	analog_audio_source=dmix
serial_param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200,	serial_param=9600-8n1

Parameter	Description	Example
	38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	
edid	EDID content.	Read only
sinkpower.mode	Three modes: ALL, CEC, RS232.	sinkpower.mode=ALL
sinkpower.cec.onetouchplay	The value of “OneTouchPlay” controlled by CEC, if there’re combined CEC commands, each two different commands would be connected by a commas (,).	sinkpower.cec.onetouchplay=4004
sinkpower.cec.standby	The value of “StandBy” controlled by CEC, if there’re combined CEC commands, each two different commands would be connected by a commas (,).	sinkpower.cec.standby=ff36
sinkpower.rs232.cmdmode	RS232 control mode, includes hex and ascii.	sinkpower.rs232.cmdmode=ascii
sinkpower.rs232.param	RS232 serial parameter. The parameter includes baud rate, data bits, parity bits and stop bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	sinkpower.rs232.param=9600-8n1
sinkpower.rs232.onetouchplay	RS232 command that used to control OneTouchPlay.	sinkpower.rs232.onetouchplay= "Power on"
sinkpower.rs232.standby	RS232 command that used to control StandBy.	sinkpower.rs232.standby="Power off"

3.1.7 DIPE6000V3

Parameters for DIPE6000V3 only:

Parameter	Description	Example
video_input	True or false. Read only.	-
video_timing	The video input resolution, which is available only if the video_input is set as true. Read only.	-
video_source	The video source mode. The value of the parameter includes auto, hdmi and dp.	video_source=auto
dante_source	Dante input mode: hdmi and analog. **Note:** This Dante input mode is available for DIPE6000V3CD-003, DIPE6000V3D-003, DIPE6000V3Du-003 and DIPE6000V3CDu-003 units only.	dante_source=hdmi
hdcp14_enable	The value of the parameter includes true and false. • true: enable HDCP 1.4. • false: disable HDCP 1.4.	hdcp14_enable=true
hdcp22_enable	The value of the parameter includes true and false. • true: enable HDCP 2.2. • false: disable HDCP 2.2.	hdcp22_enable=true
analog_audio_direction	Analog audio direction. The value of the parameter includes input and output.	analog_audio_direction=input
serial_param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate:	serial_param=9600-8n1

Parameter	Description	Example
	150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	
edid	EDID content	edid= 00ffffffffffff000614000156524c 420816010380341d780a01c1a 057479827124c4c2108008140 010101010101010101010101010 101023a801871382d40582c46 0040846300001e08e80030f270 5a80b0588a00ba892100001e0 00000fc004170746f566973696f 6e20200a000000fd0018550e85 3c000a2020202020200120020 340735801020304511121314 1f 2021220607905d5e5f6263646 061230d0707830f00006d030c0 01000383c20406801020367d8 5dc401788003e40f0000c0011d 00bc52d01e20b828554040846 300001e023a80d072382d4010 2c458040846300001e023a801 871382d40582c450040846300 001e00000000000000000f2
stream0fps_by2_enable	To enable/disable stream0 fps divided by 2. • true: enable stream0 fps divided by 2. • false: disable stream0 fps divided by 2.	stream0fps_by2_enable=false
stream0_enable	To enable/disable stream0. • true: enable stream0. • false: disable stream0.	stream0_enable=false
stream1_scale	Scale the stream1's resolution. The width must be multiples of 32.	stream1_scale=128x128
stream1fps_by2_enable	To enable/disable stream1 fps divided by 2. • true: enable stream1 fps divided by 2. • false: disable stream1 fps divided by 2.	stream1fps_by2_enable=false
stream1_enable	To enable/disable stream1. • true: enable stream1. • false: disable stream1.	stream1_enable=false
bandwidth_adjust_mode	The mode includes {0|1|2}. • 0: Main stream first, at no time would main stream be adjusted. • 1: Equilibrium mode, main stream's framerate is adjustable. • 2: Multiview stream first, if TX output bandwidth exceeds that supported after adjust main stream's frame rate, main stream can be closed.	bandwidth_adjust_mode=0
icronusb.ip.mode	static/hdcp	icronusb.ip.mode=dhcp
icronusb.ip.addr	like 192.168.11.243	icronusb.ip.addr=192.168.11.243
icronusb.ip.netmask	like 255.255.0.0	netmask=255.255.0.0
icronusb.ip.gateway	like 192.168.11.1	gateway=192.168.11.1

3.1.8 DIPE6000L(F)

Parameters for DIPE6000L&DIPE6000LF only:

Parameter	Description	Example
video_input	True or false.	Read only.
video_timing	The video input resolution, which is available only if the video_input is set as true.	Read only.
video_source	The video source mode. The value of the parameter is hdmi.	video_source=hdmi
hdcp14_enable	The value of the parameter includes true and false. • true: enable HDCP 1.4. • false: disable HDCP 1.4.	hdcp14_enable=true
hdcp22_enable	The value of the parameter includes true and false. • true: enable HDCP 2.2. • false: disable HDCP 2.2.	hdcp22_enable=true
analog_audio_direction	Analog audio direction. The value of the parameter includes input and output.	analog_audio_direction=input
serial_param	Serial parameter. The parameter includes baud rate, data bits and parity bits. Value of the baud rate: 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Value of the data bits: 5, 6, 7, 8. Value of the parity bits: o, e, n. Value of the stop bits: 1, 2.	serial_param=9600-8n1
edid	EDID content.	edid= 00ffffffffffff000614000156524c 420816010380341d780a01c1a 057479827124c4c2108008140 010101010101010101010101010 101023a801871382d40582c46 0040846300001e08e80030f270 5a80b0588a00ba892100001e0 00000fc004170746f566973696f 6e20200a000000fd0018550e85 3c000a2020202020200120020 340735801020304511121314 1f 2021220607905d5e5f6263646 061230d0707830f00006d030c0 01000383c2040680102036 7d8 5dc401788003e40f0000c0011d 00bc52d01e20b828554040846 300001e023a80d072382d4010 2c458040846300001e023a801 871382d40582c450040846300 001e00000000000000000f2
stream0_enable	To enable/disable stream0. • true: enable stream0. • false: disable stream0.	stream0_enable=false

3.1.9 IPT6000

Parameters for IPT6000 only:

Parameter	Description	Example
video_mode	Video output mode: genlock, fast_switch, genlock_scaling	video_mode=genlock
video_timing	Video output resolution, which is available only if the video_mode is set as fast_switch. Value of the resolution: AUTO, 720P@50, 720P@60, 1080P@50, 1080P@60, 2160P@25, 2160P@30, 2160P@50, 2160P@60, 4096x2160@24, 4096x2160@30, 4096x2160@60, 1024x768@60, 1280x768@60, 1280x960@60, 1280x1024@60, 1360x768@60, 1400x1050@60, 1600x1200@60, 1680x1050@60, 1920x1200@60	video_timing=720P@50
genlock_scaling_resolution	Resolution setting in genlock_scaling mode. Value of the resolution: AUTO, 720P, 1080P, 2160P, 4096x2160, 1024x768, 1280x768, 1280x960, 1280x1024, 1360x768, 1400x1050, 1600x1200, 1680x1050, 1920x1200	genlock_scaling_resolution=AUTO
video_stretch_type	Video stretch mode, which is available only if the video_mode is set as fast_switch. Value of the video stretch: none, fit_in, stretch_out	video_stretch_type=fit_in
hdmi_audio_source	HDMI audio source: hdmi, analog, dmix	hdmi_audio_source=hdmi
analog_audio_source	Analog audio source: analog, dmix	analog_audio_source=dmix
serial_param	Serial parameters, including Baud rate, Data bits, Parity, Stop bits. Example: 115200-8n1. • Value of baudrate: [150|200|300|600|1200|1800|2400|4800|9600|19200|38400|57600|115200]; • Value of data bits: [5|6|7|8]; • Value of Parity: [o|e|n]; • Value of Stop bits: [1|2].	serial_param=9600-8n1
sinkpower.mode	SinkPower mode: ALL, CEC, RS232	sinkpower.mode=ALL
sinkpower.cec.onetouchplay	The value of controlling OneTouchPlay through CEC. If multiple CEC commands are in use, use the comma "," to connect each command.	sinkpower.cec.onetouchplay=4004
sinkpower.cec.standby	The value of controlling Standby through CEC. If multiple CEC commands are in use, use the comma "," to connect each command.	sinkpower.cec.standby=ff36
sinkpower.rs232.cmdmode	The RS232 command type, including hex and ascii.	sinkpower.rs232.cmdmode=ascii
sinkpower.rs232.param	RS232 serial port parameters, including Baud rate, Data bits, Parity and Stop bits. e.g. 115200-8n1. • Value of baudrate: [150|200|300|600|1200|1800|2400|4800|9600|19200|38400|57600|115200]; • Value of data bits: [5|6|7|8]; • Value of Parity: [o|e|n]; • Value of Stop bits: [1|2].	sinkpower.rs232.param=9600-8n1
sinkpower.rs232.onetouchplay	The RS232 command to control OneTouchPlay	sinkpower.rs232.onetouchplay = "Power on"
sinkpower.rs232.standby	The RS232 command to control StandBy.	sinkpower.rs232.standby="Power off"
video_input	true, false	Read only
video_timing	The current input video resolution, available only if the video_input is set as true.	Read only

Parameter	Description	Example
video_source	Video input mode: auto, hdmi, dp	video_source=auto
dante_source	Dante input mode: hdmi, analog. This is available for DIPE6000V3D only.	dante_source=hdmi
hdcp14_enable	• true: enable HDCP 1.4. • false: disable HDCP 1.4.	hdcp14_enable=true
hdcp22_enable	• true: enable HDCP 2.2. • false: disable HDCP 2.2.	hdcp22_enable=true
analog_audio_direction	Analog audio input direction: input and output	analog_audio_direction=input
edid	EDID content	edid= 00ffffffffffff000614000156524c 420816010380341d780a01c1a 057479827124c4c2108008140 010101010101010101010101010 101023a801871382d40582c46 0040846300001e08e80030f270 5a80b0588a00ba892100001e0 00000fc004170746f566973696 f6e20200a000000fd0018550e8 53c000a202020202020012002 034073580102030451112131 4 1f2021220607905d5e5f626364 6061230d0707830f00006d030c 001000383c204068010203 67d 85dc401788003e40f0000c0011 d00bc52d01e20b82855404084 6300001e023a80d072382d401 02c458040846300001e023a80 1871382d40582c45004084630 0001e000000000000000f2
stream0fps_by2_enable	To enable/disable stream0 fps divided by 2. • true: enable stream0 fps divided by 2. • false: disable stream0 fps divided by 2.	stream0fps_by2_enable=false
stream0_enable	To enable/disable stream0. • true: enable stream0. • false: disable stream0.	stream0_enable=false
stream1_scale	Scale the stream1's resolution. The width must be multiples of 32.	stream1_scale=128x128
stream1fps_by2_enable	To enable/disable stream1 fps divided by 2. • true: enable stream1 fps divided by 2. • false: disable stream1 fps divided by 2.	stream1fps_by2_enable=false
stream1_enable	To enable/disable stream1. • true: enable stream1. • false: disable stream1.	stream1_enable=false
bandwidth_adjust_mode	The mode includes {0|1|2}. • 0: Main stream first, at no time would main stream be adjusted. • 1: Equilibrium mode, main stream's framerate is adjustable. • 2: Multiview stream first, if TX output bandwidth exceeds that supported after adjust main stream's frame rate, main stream can be closed.	bandwidth_adjust_mode=0
icronusb.ip.mode	static/hdcp	icronusb.ip.mode=dhcp
icronusb.ip.addr	like 192.168.11.243	icronusb.ip.addr=192.168.11.243
icronusb.ip.netmask	like 255.255.0.0	netmask=255.255.0.0
icronusb.ip.gateway	like 192.168.11.1	gateway=192.168.11.1
mcu.hdmi_local_loop	HDMI loopout status:	mcu.hdmi_local_loop=on

Parameter	Description	Example
	• on- enable HDMI loopout • off-disable HDMI loopout	
mcu.network_isolation_status	Dante stream isolation status: non: No Dante stream isolation. • on: enable Dante stream isolation. • off: disable Dante stream isolation.	mcu.network_isolation_status=on
mcu.usb_config (read only)	None, HID, Icron,Icron_lex, icron_rex	
mcu.usb_mode	Working mode of USB extension module: auto, lex and rex.	mcu.usb_mode=auto
mcu.usb_role (read only)	Working status of USB extension module: lex and rex.	mcu.usb_role=rex
mcu.ip.mode	static/hdcp	mcu.ip.mode=dhcp
mcu.ip.addr	like 192.168.11.243	mcu.ip.addr=192.168.11.243
mcu.ip.netmask	like 255.255.0.0	netmask=255.255.0.0
mcu.ip.gateway	like 192.168.11.1	gateway=192.168.11.1

4. Appendix

5. FAQs

1. **If errors occur when executing commands, what response will controller give?**

   In fact, responses returned by IP controller are nearly a confirmation of API commands sent from a third party control device such as a computer. Despite IP controller checked command format basically, the response isn't the actual execution result. It means that IP controller may return normal response even if errors occur in execution. Therefore, a third party control device should not use the response as the basis to judge whether a command is executed successfully, and should use the right query commands to get the system's running status to make right judgment.

2. **How can I set IP mode of TX/RX?**

   By default, TX/RX runs in AutoIP mode. You can use API command `config set device ip` to change the ir IP mode to DHCP or Static IP. For more information, see 2.1.10config set device ip. If you want to obtain TX/RX's IP mode, you can use API command `config get device info`.

3. **How can I assign friendly names (alias) to TX or RX?**

   You can use API command `config set device alias` to do this. For example, if you want assign alias **mydvd** to TX DIPE5100-002C8D123456, use `config set device alias DIPE5100-002C8D123456 mydvd`

4. **When I send API commands, how do I specify TX and RX?**

   By alias or hostname (device name). Alias and hostname are unique.

5. **What standard do the API commands use?**

   API commands are printable ASCII characters and are terminated with a <CR>, meaning a carriage return and a line feed must be followed in the end of a command.

6. **It looks like to create a video wall I would use the command "add vw-name". Once a video wall is created, how do I turn it on and off? I should be able to create multiple video wall configurations and then recall a configuration? Is this possible?**

   I would like to use the PC software to create a video wall configuration and then save the configuration as a video wall name. I would then send a telnet command to recall a video wall name. This command could be "set vw-name".

7. Except **vw** add and vw rm, other commands of vw are effective instantly. (The screen would change based on the commands). To create and store multiple video-wall configuration, or recall the configuration effective, would be depending on your 3-rd party software. Any 3-rd party software could recall a specific configuration, based on this API protocol and repeat corresponding add commands.

8. **How can I create a 2 x 2 video wall?**

   Before you use video wall, you'd better assign an alias to each device for easy management. For example, if you have 4 TX and 4 RX, do as follows.

   ```
   config set device alias DIPE5100-AAAAAAAAAAAA pc1
   config set device alias DIPE5100-BBBBBBBBBBBB pc2
   ```

```
config set device alias DIPE5100-CCCCCCCCCCCC dvd

config set device alias DIPE5100-DDDDDDDDDD stb

config set device aliasDIPD-5100-EEEEEEEEEEEEEE TopLeft

config set device aliasDIPD-5100-FFFFFFFFFFFFFFFF TopRight

config set device aliasDIPD-5100-GGGGGGGGGGG BottomLeft

config set device aliasDIPD-5100-HHHHHHHHHH BottomRight
```

You can use two methods to create a 2 x 2 video wall:

Method 1:

1. Use **vw add vw1 2 2pc1**. This command is used to create a video wall **vw1**with two rows and two columns and assign TX **pc1**.
2. Use **vw add vw1 TopLeft 1 1 TopRight 1 2 BottomLeft 2 1 BottomRight 2 2**. This command is used to add RX to video wall **vw1** and assign their positions. Once this command is executed, RX will play video wall.

Method 2:

Use **vw add vw1 layout 2 2 pc1TopLeft TopRight BottomLeft BottomRight**. This command is an easier way to add a video wall. It just needs one line of command.