API Commands for OIP-A60C

Document Version: 1.1.0

Applies to firmware: 1.6.10 or later.

Contents

1 Introduction 5
1.1 Important Notes 5
1.2 Preparation 5
1.2.1 Setting IP Address in Your Computer 5
1.2.2 Enabling Telnet Client 6
1.3 Logging In to IP Controller via Command-line Interface 6
1.4 Introduction to Terminology 7
1.5 API Commands Overview 7
2 Command Sets 8
2.1 config Commands 8
2.1.1 config get ipsetting 8
2.1.2 config get ipsetting2 8
2.1.3 config get version 9
2.1.4 configure get system info 9
2.1.5 config get controller info 10
2.1.6 config get system temperature 11
2.1.7 config get system upgrade_state 11
2.1.8 config get system last_upgrade_result 11
2.1.9 config set ip4addr2 12
2.1.10 config set newip4addr 12
2.1.11 config get newip4addr 13
2.1.12 config set dnsserver ip4addr 13
2.1.13 config get dnsserver ip4addr 13
2.1.14 config set dnsserver ip4addr2 14
2.1.15 config get dnsserver ip4addr2 14
2.1.16 config set telnetpasswd 14
2.1.17 config set delete telnetpasswd 14
2.1.18 config set system service {telnet_api|ssh_api|http|https|telnettls_api} 15
2.1.19 config set system service {telnet_api|ssh_api|telnettls_api} port 15
2.1.20 config get system service port 15
2.1.21 config set system service {ssh_api|telnettls_api} password 16
2.1.22 config set system realtime 16
2.1.23 config get system realtime 16
2.1.24 config set system lcd {ipversion|off} 17
2.1.25 config get system lcd 17
2.1.26 config get system edid_info 18
2.1.27 config set system custom_edid 18
2.1.28 config set system remove_custom_edid 18
2.1.29 config set system web_inactive_time 19
2.1.30 config get system web_inactive_time 19
2.1.31 config set system web_max_lifetime 20
2.1.32 config get system web_max_lifetime 20
2.1.33 config set system notify_enable 20
2.1.34 config get system notify_enable 21
2.1.35 config set system ntpenable 21
2.1.36 config get system ntpenable 21
2.1.37 config set system ntpserver 22
2.1.38 config get system ntpserver 22
2.1.39 config set system ntpzone 22
2.1.40 config get system ntpzone 23
2.1.41 config set system syncntp 23
2.1.42 config get system ntpserverstatus 23
2.1.43 config set system 802_1x 24
2.1.44 config get system 802_1x 24
2.1.45 config set restorefactory 24
2.1.46 config set reboot 25
2.1.47 config set device alias 25
2.1.48 config set device remove 25
2.1.49 config set device ip 26
2.1.50 config set device danteip 26
2.1.51 config set device reboot 27
2.1.52 config set device restorefactory 27
2.1.53 config set device info 28
2.1.54 config set device group update 28
2.1.55 config set device service 29
2.1.56 config set device fastswitch 29
2.1.57 config set device cec standby 29
2.1.58 config set device cec onetouchplay 30

2.1.59 config set device sinkpower....30
2.1.60 config set device osd turnon....30
2.1.61 config set device osd turnoff....31
2.1.62 config set device findme....31
2.1.63 config set device audio volume....31
2.1.64 config set session alias....32
2.1.65 config set telnet alias....32
2.1.66 config get telnet alias....32
2.1.67 config set rs-232 alias....33
2.1.68 config get rs-232 alias....33
2.1.69 config set device rs232_passthrough....33
2.1.70 config set device rs232_proxy....34
2.1.71 config set device rs232_feedback....34
2.1.72 config set device preview_params....34
2.1.73 config set device subscribe/unsubscribe....35
2.1.74 config set device multicast....36
2.1.75 config set device aes67_info....36
2.1.76 config set device dns....37
2.1.77 config get devicelist....37
2.1.78 config get name....37
2.1.79 config get device info....38
2.1.80 config get device status....42
2.1.81 config get device runtime....43
2.1.82 config get devicejsonstring....44
2.1.83 config get scenejsonstring....45
2.2 matrix Commands....45
2.2.1 matrix set....45
2.2.2 matrix get....46
2.2.3 matrix video set....47
2.2.4 matrix video get....47
2.2.5 matrix audio set....48
2.2.6 matrix audio get....48
2.2.7 matrix usb set....49
2.2.8 matrix usb get....49
2.2.9 matrix audio3 set....50
2.2.10 matrix audio3 get....50
2.2.11 matrix infrared set....51
2.2.12 matrix infrared get....51
2.2.13 matrix serial set....52
2.2.14 matrix serial get....52
2.3 scene Commands....53
2.3.1 scene get....53
2.3.2 scene active....53
2.3.3 scene set....53
2.3.4 scene change scenename txname....54
2.3.5 scene set bezelgap....54
2.3.6 scene set stretch....54
2.3.7 scene connect scenename....55
2.3.8 scene create....55
2.3.9 scene update....56
2.3.10 scene modify name....57
2.3.11 scene remove....57
2.3.12 scene group update....57
2.4 serial Commands....58
2.5 notify Commands....58
2.5.1 notify endpoint....58
2.5.2 notify serialinfo....59
2.5.3 notify device settings_changed....59
2.5.4 notify device status_changed....59
2.5.5 notify video....60
2.5.6 notify sink....60
2.5.7 notify update....60
2.5.8 notify ntpserver is {reachable | unreachable}....61
2.6 Infrared Command....61
2.6.1 Infrared Command....61
2.7 cec Commands....61
3 Concurrent Commands....62
4 Data Structures....62
4.1 Scene Data....62
4.2 Device Info....64

4.2.1 Decoder — OIP-A60D 65
4.2.2 Encoder — OIP-A60E 67

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.

1.1 Important Notes

1. **Alias based programming mechanism**

Since the API version 1.7, this mechanism is supported. This way you needn't 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 strategy change**

Since the version API V1.17/Controller Web_v8.2.3/gbcmd_v8.2.6, the gateway setting strategy has been changed to single gateway pattern, 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 using keywords:

Commands
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 get device info
config get device status
serial
infrared
cec

1.2 Preparation

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

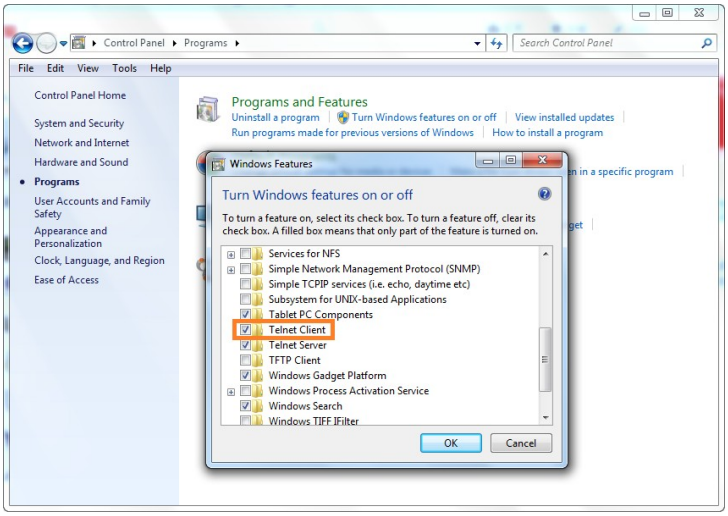
1.2.1 Setting IP Address in Your Computer

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.x.x range with a subnet mask of 255.255.0.0

1.2.2 Enabling Telnet Client

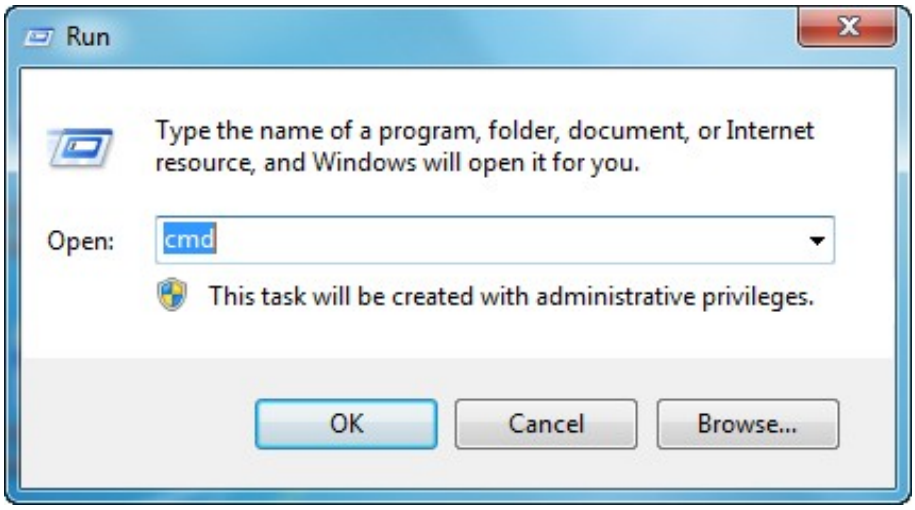
Before logging in to IP controller via command-line interface, make sure that Telnet Client is enabled. By default, Telnet Client is disabled in Windows 7. To turn on Telnet Client, do as follows.

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

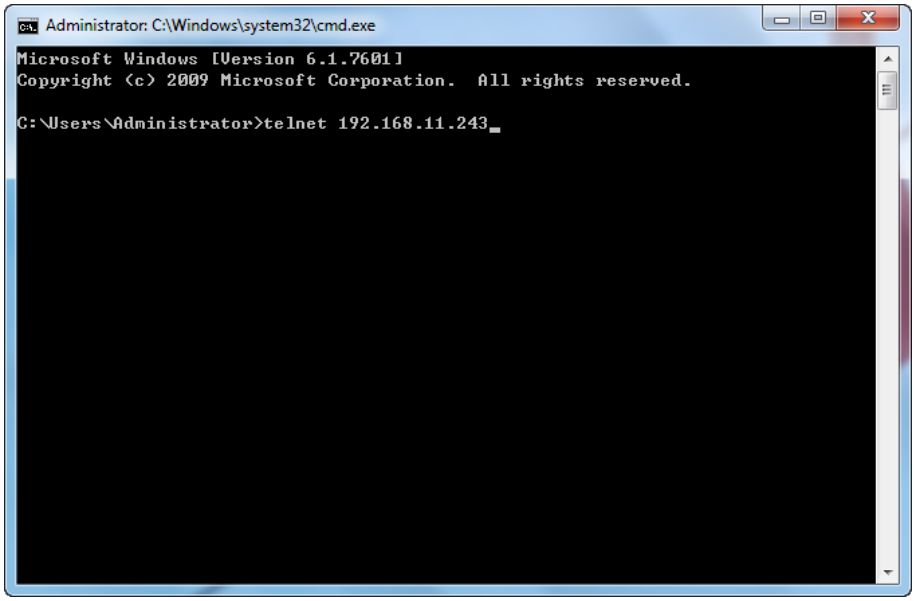


1.3 Logging In to IP Controller via Command-line Interface

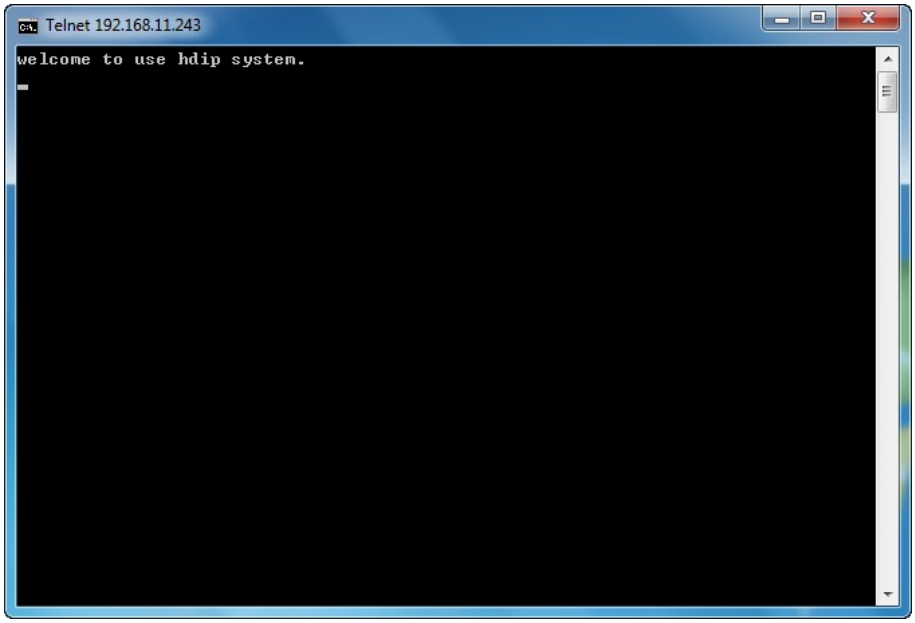
1. Choose **Start** > **Run**.
2. In **Run**, enter **cmd**, then click **OK**.



3. Enter **telnet 192.168.11.243** (the controller LAN(C) IP address), then press **Enter**.



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



1.4 Introduction to Terminology

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

Terminology	Description
Device	An endpoint device (TX or RX) controlled and managed by the 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 the illustrative hostname OIP-A60E-DCE2AC000001.
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 config set device alias.

1.5 API Commands Overview

API commands of the IP controller are categorized into the following functional groups:

Command Group	Description
2.1 config Commands	manages and configures IP controller and connected devices
2.2 matrix Commands	controls the switching of TX and RX or retrieves matrix information
2.3 scene Commands	configures and manages video wall functionality
2.4 serial Commands	sends commands to external devices via device serial interfaces
2.5 notify Commands	notifies third party control systems (e.g., PC) of serial responses and device status.
2.6 Infrared Command	transmits infrared control data through supported endpoint devices
2.7 cec Commands	transmits HDMI-CEC control data through supported endpoint devices

2 Command Sets

2.1 config Commands

2.1.1 config get ipsetting

Command	`config get ipsetting`
Response	`ipsetting is:ip4addr <ipv4_address> netmask <subnet_mask> gateway <gateway>`
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
```

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.

2.1.2 config get ipsetting2

Command	`config get ipsetting2`
Response	`ipsetting2 is:ip4addr <ipv4_address> netmask <subnet_mask> gateway <gateway>`
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
```

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.

2.1.3 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 - Service version

Example:

To obtain IP controller version information:

Command:

```
config get version
```

Response:

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

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

2.1.4 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
    }
}
```

2.1.5 config get controller info

Command	config get controller info

Response	controller info:
	[
	{
	"hostname_av": "string",
	"hostname_ctl": "string",
	"mac_av": "string",
	"mac_ctl": "string",
	"serialNumber": "string",
	"version": "string"
	}
	]

Description	Retrieves controller information, including AV hostname, control hostname, MAC addresses, serial number, and firmware version.

Example:

Command:

```
config get controller info
```

Response:

```
controller info:
[
    {
        "hostname_av": "<actual controller AV hostname>",
        "hostname_ctl": "<actual controller control hostname>",
        "mac_av": "DC:E2:AC:00:00:01",
        "mac_ctl": "DC:E2:AC:00:00:02",
        "serialNumber": "12345678912345",
        "version": "v1.2.5:AVITX_1.1"
    }
]
```

Content enclosed in angle brackets is a placeholder, not an actual returned value.

MAC addresses in this illustration are illustrative example values, not actual device addresses.

2.1.6 config get system temperature

Command	`config get system temperature`
Response	`System Temperature: {temperature}`
Description	Queries the controller's current temperature. Note: • If this command is unavailable in the selected API version, the controller returns UNKNOWN COMMAND.

Example:

To query the controller's current temperature:

Command:

```
config get system temperature
```

Response:

```
System Temperature:48.365
```

2.1.7 config get system upgrade_state

Command	`config get system upgrade_state`
Response	`system upgrade_state is {idle | transferring | upgrading | completed | background_completed}`
Description	Queries the current system upgrade status. idle – The system is idle. transferring – The upgrade file is being transferred. upgrading – The system is being upgraded. completed – The upgrade process is complete. background_completed – The previous upgrade failed or was interrupted, and the subsequent background upgrade has completed successfully.

Example:

Command:

```
config get system upgrade_state
```

Response:

```
system upgrade_state is idle
```

2.1.8 config get system last_upgrade_result

Command	`config get system last_upgrade_result`
Response	`system last_upgrade_result is {failure | success | idle}`
Description	Queries the result of the previous system upgrade. idle – The system has never been upgraded. failure – The previous upgrade failed. success – The previous upgrade completed successfully.

Example:

Command:

```
config get system last_upgrade_result
```

Response:

```
system last_upgrade_result is success
```

2.1.9 config set ip4addr2

Command	`config set ip4addr2 <ipv4_address> netmask <ipv4_address> gateway <ipv4_address>`
Response	`ip2 setting will change to: ipaddr <ipv4_address> netmask <ipv4_address> gateway <ipv4_address>`
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
```

2.1.10 config set newip4addr

Command	`config set newip4addr {dhcp|static} ip4addr netmask gateway`
Response	`ip setting will change to: dhcp/static ip4addr netmask gateway`
Description	Configures the network settings of the controller's primary AV port used for communication with TX/RX devices. Note: Supports both DHCP and Static IP modes. In Static mode, the IP address, subnet mask, and gateway may be configured individually or together. Network settings take effect after reboot.

Example 1:

To configure the AV port to obtain an IP address automatically:

Command:

```
config set newip4addr dhcp
```

Response:

```
ip setting will change to: dhcp
```

Example 2:

To configure a static IP address for the AV port:

Command:

```
config set newip4addr static 169.254.1.1 255.255.0.0 169.254.2.1
```

Response:

```
ip setting will change to: static 169.254.1.1 255.255.0.0 169.254.2.1
```

2.1.11 config get newip4addr

Command	config get newip4addr
Response	ipsetting is:dhcp/static <ipv4_address> <ipv4_address> <ipv4_address>
Description	Retrieves the network settings of the controller's primary AV port used for communication with TX/RX devices.

Example:

To retrieve the network settings of the AV port:

Command:

```
config get newip4addr
```

Response:

```
ipsetting is:static 169.254.1.100 255.255.0.0 169.254.1.1
```

Or:

```
ipsetting is:dhcp 169.254.1.100 255.255.0.0 169.254.1.1
```

2.1.12 config set dnsserver ip4addr

Command	config set dnsserver ip4addr [IPv4 address]
Response	dns server ip setting will change to: [IPv4 address]
Description	Sets the primary DNS server IPv4 address for the controller. The configuration takes effect immediately after the command is received.

Example:

Command:

```
config set dnsserver ip4addr 202.96.134.133
```

Response:

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

2.1.13 config get dnsserver ip4addr

Command	config get dnsserver ip4addr
Response	dns server ip4addr is: [IPv4 address]
Description	Retrieves the primary DNS server IPv4 address configured on the controller.

Example:

Command:

```
config get dnsserver ip4addr
```

Response:

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

2.1.14 config set dnsserver ip4addr2

Command	`config set dnsserver ip4addr2 [IPv4 address]`
Response	`dns server ip2 setting will change to: [IPv4 address]`
Description	Sets the secondary DNS server IPv4 address for the controller. The configuration takes effect immediately after the command is received.

Example:

Command:

```
config set dnsserver ip4addr2 202.96.128.68
```

Response:

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

2.1.15 config get dnsserver ip4addr2

Command	`config get dnsserver ip4addr2`
Response	`dns server ip4addr2 is: [IPv4 address]`
Description	Retrieves the secondary DNS server IPv4 address for the controller.

Example:

Command:

```
config get dnsserver ip4addr2
```

Response:

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

2.1.16 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
```

2.1.17 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
```

2.1.18 config set system service {telnet_api|ssh_api|http|https|telnettls_api}

Command	`config set system service {telnet_api|ssh_api|http|https|telnettls_api} {on|off}`
Response	`config set system service {telnet_api|ssh_api|http|https|telnettls_api} {on|off}`
Description	Enables or disables the specified system service.

Example:

To enable the HTTPS service:

Command:

```
config set system service https on
```

Response:

```
config set system service https on
```

2.1.19 config set system service {telnet_api|ssh_api|telnettls_api} port

Command	`config set system service {telnet_api|ssh_api|telnettls_api} port n`
Response	`config set system service {telnet_api|ssh_api|telnettls_api} port n`
Description	Configures the port number for the specified system service.

Example:

To configure the telnet_api service to use port 23:

Command:

```
config set system service telnet_api port 23
```

Response:

```
config set system service telnet_api port 23
```

2.1.20 config get system service port

Command	`config get system service {telnet_api|ssh_api|telnettls_api} port`
Response	`system service {telnet_api|ssh_api|telnettls_api} port is n`
Description	Queries the port number configured for the specified external system service.

Example:

To query the port used by the telnet_api service:

Command:

```
config get system service telnet_api port
```

Response:

```
system service telnet_api port is 23
```

2.1.21 config set system service {ssh_api|telnettls_api} password

Command	`config set system service {ssh_api|telnettls_api} password <old_password> <new_ password> <confirm_password>`
Response	`config set system service {ssh_api|telnettls_api} password success`
Description	Changes the login password for the specified service.

Example:

To change ssh_api service password from “admin” to “123456”:

Command:

```
config set system service ssh_api password admin 123456 123456
```

Response:

```
config set system service ssh_api password success
```

or:

```
ssh_api current password error
```

2.1.22 config set system realtime

Command	`config set system realtime <DD-MM-YYYY> <HH:MM:SS>`
Response	`config set system realtime <DD-MM-YYYY> <HH:MM:SS>`
Description	Configures the RTU real-time parameters. DD-MM-YYYY – Specifies the date. HH:MM:SS – Specifies the time.
Error	Returns UNKNOWN COMMAND if the specified API version does not support this command.

Example:

Command:

```
config set system realtime 30-06-2026 14:27:50
```

Response:

```
config set system realtime 30-06-2026 14:27:50
```

2.1.23 config get system realtime

Command	`config get system realtime`
Response	`RealTime:<weekday>,<DD-MM-YYYY>,<HH:MM:SS>`
Description	Retrieves the RTU real-time parameters.

Example:

Command:

```
config get system realtime
```

Response:

```
RealTime:Tue,30-06-2026,14:27:50
```

2.1.24 config set system lcd {ipversion|off}

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

Example:

To enable the LCD screen to display the controller IP address and API version:

Command:

```
config set system lcd ipversion
```

Response:

```
config set system lcd ipversion
```

2.1.25 config get system lcd

Command	`config get system lcd`
Response	`config get system lcd {ipversion|off}`
Description	Retrieves the LCD display configuration of the controller. ipversion: Displays the IP address and firmware version. off: Turns off the LCD display. Default: ipversion

Example:

To retrieve the LCD display configuration of the controller:

Command:

```
config get system lcd
```

Response:

```
config get system lcd ipversion
```

2.1.26 config get system edid_info

Command	`config get system edid_info`
Response	`system edid_info is: { "mjpeg": [ { "edid_hex": "...", "edid_name": "2560x1440_2D_2CH.bin", "type": "buildin" }, ... ] }`
Description	Queries the built-in and custom EDID information.

Example

To query the EDID information:

Command:

```
config get system edid_info
```

Response:

```
system edid_info is: {
    "mjpeg": [
        {
            "edid_hex": "...",
            "edid_name": "2560x1440_2D_2CH.bin",
            "type": "buildin"
        },
        ...
    ]
}
```

2.1.27 config set system custom_edid

Command	`config set system custom_edid {json}`
Response	`set custom_edid success`
Description	Adds a custom EDID. A maximum of 32 custom EDIDs can be added.

Example

To add a custom EDID:

Command:

```
config set system custom_edid [{"edid_hex":"00 ff ff ff ff ff ff 00
...","edid_name":"cus31080P60_REC709_8CH.bin","type":"mjpeg"}]
```

Response:

```
set custom_edid success
```

2.1.28 config set system remove_custom_edid

Command	`config set system remove_custom_edid [{json}]`
Response	`remove custom edid success`
Description	Removes one or more custom EDIDs. EDID file names containing spaces are supported.

Example

To remove a custom EDID:

Command:

```
config set system remove_custom_edid [{"type":"mjpeg","edid_name":"EDID
111.bin"}]
```

Response:

```
remove custom edid success
```

2.1.29 config set system web_inactive_time

Command	`config set system web_inactive_time {0|15|30|60|120|240|480|720|1440}`
Response	`set system web_inactive_time {time} success`
Description	Sets the Web UI session inactivity timeout in minutes. A Web client is disconnected if no activity is detected within the configured interval. Note: • Default value: 15 minutes. • 0 disables the inactivity timeout.

Example:

To set the Web UI session inactivity timeout to 15 minutes:

Command:

```
config set system web_inactive_time 15
```

Response:

```
set system web_inactive_time 15 success
```

2.1.30 config get system web_inactive_time

Command	`config get system web_inactive_time`
Response	`system web inactive time is {time}`
Description	Queries the configured Web UI session inactivity timeout. Note: • The returned value is expressed in minutes.

Example:

To query the Web UI session inactivity timeout:

Command:

```
config get system web_inactive_time
```

Response:

```
system web inactive time is 15
```

2.1.31 config set system web_max_lifetime

Command	`config set system web_max_lifetime {0|120|240|480|720|1440|10080}`
Response	`set system web_max_lifetime {time} success`
Description	Sets the maximum Web UI session lifetime in minutes. The corresponding Web client is disconnected when the configured lifetime is reached. Note: • Default value: 120 minutes. • 0 disables the maximum session lifetime.

Example:

To set the maximum Web UI session lifetime to 120 minutes:

Command:

```
config set system web_max_lifetime 120
```

Response:

```
set system web_max_lifetime 120 success
```

2.1.32 config get system web_max_lifetime

Command	`config get system web_max_lifetime`
Response	`system web max lifetime is {time}`
Description	Queries the configured maximum Web UI session lifetime. Note: • The returned value is expressed in minutes.

Example:

To query the maximum Web UI session lifetime:

Command:

```
config get system web_max_lifetime
```

Response:

```
system web max lifetime is 120
```

2.1.33 config set system notify_enable

Command	`config set system notify_enable {on|off}`
Response	`set system notify_enable {on|off} success`
Description	Enables or disables proactive notifications for device setting and status changes.

Example:

To enable proactive device change notifications:

Command:

```
config set system notify_enable on
```

Response:

```
set system notify_enable on success
```

2.1.34 config get system notify_enable

Command	`config get system notify_enable`
Response	`system notify_enable is {on|off}`
Description	Queries whether proactive device change notifications are enabled.

Example:

To query the proactive device change notification setting:

Command:

```
config get system notify_enable
```

Response:

```
system notify_enable is on
```

2.1.35 config set system ntpenable

Command	`config set system ntpenable {on | off}`
Response	`set system ntpenable success`
Description	Enables or disables the NTP function. on | off –Specifies whether to enable or disable the NTP function.

Example

To enable the NTP function:

Command:

```
config set system ntpenable on
```

Response:

```
set system ntpenable success
```

2.1.36 config get system ntpenable

Command	`config get system ntpenable`
Response	`system ntpenable is {on | off}`
Description	Queries the NTP function status. Default: off

Example

To query the NTP function status:

Command:

```
config get system ntpenable
```

Response:

```
system ntpenable is on
```

2.1.37 config set system ntpserver

Command	`config set system ntpserver {<ipv4_address> | <hostname>}`
Response	`set system ntpserver success`
Description	Specifies the NTP server address used for time synchronization. The NTP server address can be configured using an IP address or a domain name.

Example

To configure an NTP server:

Command:

```
config set system ntpserver 0.pool.ntp.org
```

Response:

```
set system ntpserver success
```

2.1.38 config get system ntpserver

Command	`config get system ntpserver`
Response	`system ntpserver is <ntp_server>`
Description	Queries the configured NTP server address. If no NTP server has been configured, the default NTP server address 0.pool.ntp.org is returned.

Example

To query the NTP server address:

Command:

```
config get system ntpserver
```

Response:

```
system ntpserver is 0.pool.ntp.org
```

2.1.39 config set system ntpzone

Command	`config set system ntpzone <region> <city>`
Response	`set system ntpzone success`
Description	Sets the system time zone using a region and city identifier.

Example:

To set the system time zone to Asia/Shanghai:

Command:

```
config set system ntpzone Asia Shanghai
```

Response:

```
set system ntpzone success
```

2.1.40 config get system ntpzone

Command	`config get system ntpzone`
Response	`system ntpzone is <timezone>`
Description	Queries the system time zone. Note: • If no time zone has been configured, the default time zone Etc UTC (UTC+0) is returned.

Example

To query the system time zone:

Command:

```
config get system ntpzone
```

Response:

```
system ntpzone is Asia Shanghai
```

2.1.41 config set system syncntp

Command	`config set system syncntp [{<ipv4_address> | <hostname>}]`
Response	`set system syncntp success`
Description	Synchronizes the system time with an NTP server immediately. When no server address is specified, the configured NTP server is used. When a server address is specified, the specified server is used for synchronization.

Example

To synchronize the system time with an NTP server:

Command:

```
config set system syncntp 0.pool.ntp.org
```

Response:

```
set system syncntp success
```

2.1.42 config get system ntpserverstatus

Command	`config get system ntpserverstatus`
Response	`system ntpserverstatus is {reachable | unreachable}`
Description	Queries the NTP server connection status. reachable – Indicates that the NTP server is accessible. unreachable – Indicates that the NTP server is inaccessible.

Example

To query the NTP server status:

Command:

```
config get system ntpserverstatus
```

Response:

```
system ntpserverstatus is reachable
```

2.1.43 config set system 802_1x

Command	config set system 802_1x <settings_json>
Response	System ieee802_1x setup success
Description	Configures 802.1X-related parameters. Returns “UNSUPPORTED_FEATURE” if the specified API version does not support this feature.

Example:

To configure IEEE802.1X settings:

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" } ]
```

Response:

```
System ieee802_1x setup success
```

2.1.44 config get system 802_1x

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"
	}
	]

Description	Retrieves the 802.1X configuration information of the controller.

2.1.45 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
```

2.1.46 config set reboot

Command	config set reboot
Response	system will reboot now
Description	Reboot IP controller.

Example:

If you want to reboot IP controller:

Command:

```
config set reboot
```

Response:

```
system will reboot now
```

2.1.47 config set device alias

Command	config set device alias <hostname> xxxx
Response	<hostname>’s alias is xxxx
Description	Renames device

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 set OIP-A60D-DCE2AC000101's alias as MYDVD:

Command:

```
config set device alias OIP-A60D-DCE2AC000101 MYDVD
```

Response:

```
OIP-A60D-DCE2AC000101's alias is MYDVD
```

2.1.48 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: hostname1and 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 of OIP-A60D-DCE2AC000101 and OIP-A60D-DCE2AC000102:

Command:

```
config set device remove OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102
```

Response:

```
the following device's record will be removed:
OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

2.1.49 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: hostname1and 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 set OIP-A60D-DCE2AC000102 to AutoIP and OIP-A60D-DCE2AC000101 to Static IP (IP address 169.254.5.253, subnet mask 255.255.0.0, gateway 169.254.1.253):

Command:

```
config set device ip OIP-A60D-DCE2AC000102 autoip, OIP-A60D-DCE2AC000101 static
169.254.5.253 255.255.0.0 169.254.1.253
```

Response:

```
Devices's ipsetting will change to:
OIP-A60D-DCE2AC000102 autoip
OIP-A60D-DCE2AC000101 static 169.254.5.253 255.255.0.0 169.254.1.253
```

2.1.50 config set device danteip

Applies to	OIP-A60D

Command	`config set device danteip <hostname1> {dhcp|static} ip4addr netmask gateway, <hostname2> ...`

Response	`Dante's ipsetting will change to:`
	`<hostname1> static ip4addr netmask gateway`
	`<hostname2> dhcp`

Description	Configures the Dante IP mode for the specified devices. For static mode, specify the IP address, subnet mask, and gateway after the hostname. Multiple devices can be configured in a single command, separated by commas.

For OIP-A60D devices:
When using static mode, specify the IP address and subnet mask.
The gateway parameter is reserved for API consistency and can be set to 0.0.0.0. The device does not apply the gateway setting.
Restart required for changes to take effect.

Example:

To configure Dante IP settings for multiple devices:

Command:

```
config set device danteip OIP-A60E-DCE2AC000002 dhcp, OIP-A60E-DCE2AC000001
static 169.254.5.253 255.255.0.0 169.254.1.253
```

Response:

```
Dante's ipsetting will change to:
OIP-A60E-DCE2AC000002 dhcp
OIP-A60E-DCE2AC000001 static 169.254.5.253 255.255.0.0 169.254.1.253
```

2.1.51 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 reboot OIP-A60D-DCE2AC000202 and OIP-A60D-DCE2AC000201:

Command:

```
config set device reboot OIP-A60D-DCE2AC000202 OIP-A60D-DCE2AC000201
```

Response:

```
the following device will reboot now:
OIP-A60D-DCE2AC000202
OIP-A60D-DCE2AC000201
```

2.1.52 config set device restorefactory

Command	`config set device restorefactory <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 OIP-A60D-DCE2AC000202 and OIP-A60D-DCE2AC000201 to factory defaults:

Command:

```
config set device restorefactory OIP-A60D-DCE2AC000202 OIP-A60D-DCE2AC000201
```

Response:

```
the following device will restore to factory setting now:
OIP-A60D-DCE2AC000202
OIP-A60D-DCE2AC000201
```

2.1.53 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 Device Info.

Example:

If you want to set OIP-A60D-DCE2AC000201's mic_volume as 20, audio.mic1.gain 12 and audio.lineout1.volume20:

Command:

```
config set device info mic_volume=20 audio.mic1.gain=12 audio.lineout1.volume=20
OIP-A60D-DCE2AC000201
```

Response:

```
config set device info mic_volume=20 audio.mic1.gain=12 audio.lineout1.volume=20
OIP-A60D-DCE2AC000201
```

See Device Info for field definitions and local applicability.

2.1.54 config set device group update

Command	`config set device group update devicegroupjsonstring`
Response	`device group or sort update success`
Description	Updates device group assignments and device sort order using devicegroupjsonstring. Note: • For dedicated TX or RX devices, sequence specifies the device sort order. • For transceiver (TRX) devices, sequence specifies the RX sort order and sequence4tr specifies the TX sort order. • ungrouped is the default system group and has sequence number 1. User-defined groups start at sequence number 2.

Example:

To assign a device to group1 and update its sort order:

Command:

```
config set device group update [{"group":[{"name":"group1","sequence":5}],"sequen
ce":0,"sequence4tr":0,"name":"OIP-A60D-DCE2AC000201"}]
```

Response:

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

2.1.55 config set device service

Command	`config set device service {telnet_api|ssh_api|http|https} {on|off} <hostname1> <hostname2>...`
Response	`config set device service {telnet_api|ssh_api|http|https} {on|off} <hostname1> <hostname2>...`
Description	Enables or disables the specified management service on one or more devices. Note: • hostname1 and hostname2 are device names.

Example:

To enable the HTTPS service:

Command:

```
config set device service https on
```

Response:

```
config set device service https on
```

2.1.56 config set device fastswitch

Command	`config set device fastswitch {on|off} <hostname1> <hostname2>...`
Response	`config set device fastswitch {on|off} <hostname1> <hostname2>...`
Description	Enables or disables fast switching on the specified devices. Note: • For TX encoders, both on and off are supported. • For RX decoders, only on is supported.

Example:

To enable fast switching on a device:

Command:

```
config set device fastswitch on OIP-A60D-DCE2AC000101
```

Response:

```
config set device fastswitch on OIP-A60D-DCE2AC000101
```

2.1.57 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 RX OIP-A60D-DCE2AC000101 enter standby mode:

Command:

```
config set device cec standby OIP-A60D-DCE2AC000101
```

Response:

```
config set device cec standby OIP-A60D-DCE2AC000101
```

2.1.58 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 RX OIP-A60D-DCE2AC000101:

Command:

```
config set device cec onetouchplay OIP-A60D-DCE2AC000101
```

Response:

```
config set device cec onetouchplay OIP-A60D-DCE2AC000101
```

2.1.59 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 brings it into standby mode.

Example:

If you want to wake up a display device connected to OIP-A60D-DCE2AC000101 from its standby mode:

Command:

```
config set device sinkpower on OIP-A60D-DCE2AC000101
```

Response:

```
config set device sinkpower on OIP-A60D-DCE2AC000101
```

2.1.60 config set device osd turnon

Command	`config set device osd turnon <hostname1> <hostname2> ...`
Response	`config set device osd turnon <hostname1> <hostname2> ...`
Description	Enables the OSD display for specified devices.

Example:

Command:

```
config set device osd turnon OIP-A60D-DCE2AC000101
```

Response:

```
config set device osd turnon OIP-A60D-DCE2AC000101
```

2.1.61 config set device osd turnoff

Command	`config set device osd turnoff <hostname1> <hostname2> ...`
Response	`config set device osd turnoff <hostname1> <hostname2> ...`
Description	Disables the OSD display for specified devices.

Example:

Command:

```
config set device osd turnoff OIP-A60D-DCE2AC000101
```

Response:

```
config set device osd turnoff OIP-A60D-DCE2AC000101
```

2.1.62 config set device findme

Command	`config set device findme {0-?} <hostname1> <hostname2> ...`
Response	`config set device findme {0-?} <hostname1> <hostname2> ...`
Description	Causes the status LEDs of the specified devices (hostname1, hostname2, etc.) to blink for identification purposes. A value of 0 stops blinking; any other value specifies the blinking duration. The LEDs blink at a frequency of 4 times per second.

Example:

To trigger LED blinking on a specific device:

Command:

```
config set device findme 60 OIP-A60D-DCE2AC000101
```

Response:

```
config set device findme 60 OIP-A60D-DCE2AC000101
```

2.1.63 config set device audio volume

Command	`config set device audio volume {up|down|digit} {hdmi[:n]|analog[:n]|all} <hostname1> <hostname2> ...`
Response	`config set device audio volume {up|down|digit} {hdmi[:n]|analog[:n]|all} <hostname1> <hostname2> ...`
Description	Controls device audio volume. up increases the volume, down decreases the volume, and digit sets a specific volume value. hdmi selects all HDMI audio outputs, hdmi[:n] selects HDMI audio output n, analog selects all analog audio outputs, analog[:n] selects analog audio output n, and all selects all HDMI and analog audio outputs. Note: The up and down operations apply to analog audio only.

Example:

If you want to increase all the analog outputs audio volume of OIP-A60D-DCE2AC000102 and OIP-A60D-DCE2AC000101:

Command:

```
config set device audio volume up analog OIP-A60D-DCE2AC000102
OIP-A60D-DCE2AC000101
```

Response:

```
config set device audio volume up analog OIP-A60D-DCE2AC000102
OIP-A60D-DCE2AC000101
```

2.1.64 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
```

2.1.65 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:

Command:

```
config set telnet alias off
```

Response:

```
config set telnet alias off
```

2.1.66 config get telnet alias

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

Example:

Command:

```
config get telnet alias
```

Response:

```
telnet alias is off
```

2.1.67 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
```

2.1.68 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
```

2.1.69 config set device rs232_passthrough

Command	`config set device rs232_passthrough timeout[0, 60] <hostname> ...`
Response	`config set device rs232_passthrough timeout[0, 60] <hostname> ...`
Description	Sets RS-232 to pass-through mode and specifies the timeout.

Example:

To set RS-232 to pass-through mode with a timeout of 30:

Command:

```
config set device rs232_passthrough 30 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device rs232_passthrough 30 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

2.1.70 config set device rs232_proxy

Command	config set device rs232_proxy timeout[0, 60] port[1, 65535] <hostname> ...
Response	config set device rs232_proxy timeout[0, 60] port[1, 65535] <hostname> ...
Description	Sets RS-232 to proxy mode and specifies the timeout and port.

Example:

To set RS-232 to proxy mode:

Command:

```
config set device rs232_proxy 30 36000 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device rs232_proxy 30 36000 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

2.1.71 config set device rs232_feedback

Command	config set device rs232_feedback {HEX | ASCII} test <hostname> ...
Response	config set device rs232_feedback {HEX | ASCII} test <hostname> ...
Description	Sets RS-232 to feedback mode. The data mode must be uppercase: HEX or ASCII. test is the feedback test value, and hostname specifies one or more target devices.

Example:

To set RS-232 to feedback mode:

Command:

```
config set device rs232_feedback HEX test OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device rs232_feedback HEX test OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

2.1.72 config set device preview_params

Command	config set device preview_params {on | off} <width> <height> <fps> <bandwidth> <hostname1> <hostname2> ...
Response	config set device preview_params {on | off} <width> <height> <fps> <bandwidth> <hostname1> <hostname2> ...
Description	Enables or disables preview and configures the preview width, height, frame rate, and bandwidth.

- **width**: Integer; 80 <= width <= 1280.
- **height**: The source writes {45,720}; whether this denotes a range requires confirmation.
- **fps**: Integer; 1 <= fps <= 60.
- **bandwidth**: Integer; 10 <= bandwidth <= 50000.

Example 1:

To enable preview and configure the preview parameters:

Command:

```
config set device preview_params on 1280 72 60 8000 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device preview_params on 1280 72 60 8000 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Example 2:

To disable preview:

Command:

```
config set device preview_params off OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

Response:

```
config set device preview_params off OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

2.1.73 config set device subscribe/unsubscribe

Command	`config set device {subscribe|unsubscribe} {all|subset|api} <hostname1> <hostname2>...`
Response	`config set device {subscribe|unsubscribe} {all|subset|api} <hostname1> <hostname2>...`
Description	Subscribes or unsubscribes the specified devices to or from a notification service. subscribe | unsubscribe – Specifies whether to subscribe or unsubscribe. all – Specifies all notification types. subset – Specifies one of the following notification types: settings_changed status_changed routing_changed api – Specifies an API names starting with set_. (Reserved for future extension.) Note: After subscription, the API client receives notifications when related API settings or status change.

Example 1:

To subscribe a device to status change notification.

Command:

```
config set device subscribe status_changed OIP-A60D-DCE2AC000101
```

Response:

```
config set device subscribe status_changed OIP-A60D-DCE2AC000101
```

Example 2:

To unsubscribe a device from all notification types:

Command:

```
config set device unsubscribe all OIP-A60D-DCE2AC000101
```

Response:

```
config set device unsubscribe all OIP-A60D-DCE2AC000101
```

2.1.74 config set device multicast

Command	config set device multicast {on | off} ip_prefix[224,239] ttl[1,255] <hostname> ...
Response	config set device multicast {on | off} ip_prefix[224,239] ttl[1,255] <hostname> ...
Description	Enables or disables multicast for TX devices and configures the multicast IP prefix and TTL.

Example 1

To enable multicast:

Command:

```
config set device multicast on 239 64 OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

Response:

```
config set device multicast on 239 64 OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

Example 2

To disable multicast:

Command:

```
config set device multicast off OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

Response:

```
config set device multicast off OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

2.1.75 config set device aes67_info

Command	config set device aes67_info session_name ip port <hostname> ...
Response	config set device aes67_info session_name ip port <hostname> ...
Description	Configures the AES67 TX session name, IP address, and port. Note: The session name must contain 1–64 characters and can contain only letters, numbers, hyphens (-), and underscores (_).

Example:

To configure AES67 TX parameters:

Command:

```
config set device aes67_info myaes67 239.2.0.1 5050 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device aes67_info myaes67 239.2.0.1 5050 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

2.1.76 config set device dns

Command	`config set device dns {dns1} {dns2} <hostname> ...`
Response	`config set device dns {dns1} {dns2} <hostname> ...`
Description	Configures the primary and secondary DNS server addresses of the specified devices. Note: If a DNS server is not configured, its address is 0.0.0.0.

Example:

To configure the primary and secondary DNS server addresses:

Command:

```
config set device dns 114.114.114.114 8.8.8.8 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

Response:

```
config set device dns 114.114.114.114 8.8.8.8 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000001
```

2.1.77 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 is OIP-A60D-DCE2AC000201 OIP-A60D-DCE2AC000101
```

The current online devices are OIP-A60D-DCE2AC000201 and OIP-A60D-DCE2AC000101.

2.1.78 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 obtain OIP-A60E-DCE2AC000001's alias:

Command:

```
config get name OIP-A60E-DCE2AC000001
```

Response:

```
OIP-A60E-DCE2AC000001's alias is testEncoder
```

Example 2:

If you want to obtain OIP-A60E-DCE2AC000002's alias which is not set:

Command:

```
config get name OIP-A60E-DCE2AC000002
```

Response:

```
OIP-A60E-DCE2AC000002's alias is NULL
```

Example 3:

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

Command:

```
config get name
```

Response:

```
OIP-A60E-DCE2AC000001's alias is testEncoder
OIP-A60E-DCE2AC000002's alias is NULL
```

2.1.79 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 useKey:Value format. Key is a parameter name and value is its value. For more information, see Device Info.

Example 1:

If you want to obtain OIP-A60E-DCE2AC000001's working parameters:

Command:

```
config get device info OIP-A60E-DCE2AC000001
```

Response:

```
devices json info:
{
    "devices": [
        {
            "aliasname": "RX-1",
            "name": "OIP-A60E-DCE2AC000001",
            "version": "v2.5.8",
            "ip_mode": "dhcp",
            "ip4addr": "169.254.107.239",
            "netmask": "255.255.0.0",
            "mac": "DC:E2:AC:00:00:01",
            "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"
        }
    ]
}
```

MAC addresses in this illustration are illustrative example values, not actual device addresses.

Example2:

If you want to obtain OIP-A60D-DCE2AC000101's working parameters:

Command:

```
config get device info OIP-A60D-DCE2AC000101
```

Response:

```
devices json info:
{
    "devices": [
        {
            "name": "OIP-A60D-DCE2AC000101",
            "version": "v2.5.6",
            "ip_mode": "autoip",
            "ip4addr": "169.254.5.173",
            "netmask": "255.255.0.0",
            "mac": "DC:E2:AC:00:00:04",
            "gateway": " ",
            "hdcp": false,
            "sourcein": "null",
            "audio": [
                {
                    "name": "lineout1",
                    "mute": false
                }
            ]
        }
    ]
}
```

MAC addresses in this illustration are illustrative example values, not actual device addresses.

Example 3:

If you want to obtain the working parameters of OIP-A60E-DCE2AC000002:

Command:

```
config get device info OIP-A60E-DCE2AC000002
```

Response:


```
 {
    "aliasname": "OIP-A60E-DCE2AC000006",
    "analog_audio_direction": "INPUT",
    "bit_perpixel": 8,
    "color_space": "RGB",
    "edid": "00ffffffffffff0010a6000156524c420816010380341d780a01c1a057479827124c
4c21080081400101010101010101010101010101023a801871382d40582c460040846300001e08e80
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": "OIP-A60E-DCE2AC000002",
    "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"
}
```


See Device Info for field definitions and local applicability.

2.1.80 config get device status

Command	`config get device status <hostname1> <hostname2> ...`

Response	Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.
	devices status json info:
	{
	"devices_status": [
	{
	"aliasname": "TX-1",
	"name": "OIP-A60E-DCE2AC000001",
	" 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 OIP-A60E-DCE2AC000001's status:

Command:

```
config get device status OIP-A60E-DCE2AC000001
```

Response:

```
devices status json info:
{
    "devices_status": [
        {
            "aliasname": "TX-1",
            "name": "OIP-A60E-DCE2AC000001",
            " 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",
        }
    ]
}
```

2.1.81 config get device runtime

Command	`config get device runtime <hostname1> <hostname2> ...`

Response	`Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.`
	`devices runtime info:`
	`{`
	`"devices runtime": [`
	`{`
	`"aliasname": "",`
	`"error": "no such device",`
	`"name": "OIP-A60D-DCE2AC000201"`
	`},`
	`{`
	`"aliasname": "",`
	`"name": "OIP-A60E-DCE2AC000001",`
	`"runtime": "22541.78"`
	`}`
	`]`
	`}`

Description	Retrieves the runtime information of the specified devices. The runtime information is returned in JSON format. The runtime field is reported by each device, and the controller transparently forwards the information without modification.

Example:

To query the runtime information of specified devices:

Command:

```
config get device runtime OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000201
```

Response:

```
Same as the Response section above.
```

2.1.82 config get devicejsonstring

Command	`config get devicejsonstring`

Response	`device json string: [`
	`{`
	`"aliasName": "xxx",`
	`"deviceType": "Transmitter/Receiver",`
	`"group": [`
	`{`
	`"name": "xxx",`
	`"sequence": xxx`
	`}`
	`],`
	`"ip": "<ipv4_address>",`
	`"online": true/false,`
	`"sequence": xxx,`
	`"trueName": "xxx"`
	`}…`
	`]`

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": "OIP-A60E-DCE2AC000001"
    }…
]
```

2.1.83 config get scenejsonstring

Command	`config get scenejsonstring`
Response	`<scene_json>`
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 (Controller)."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.

See Scene Data for the complete JSON structure.

2.2 matrix Commands

2.2.1 matrix set

Command	`matrix set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Response	`matrix set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Description	Controls the switching of RX to TX. Parameters are separated by commas such as segments OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102, OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104. 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 OIP-A60E-DCE2AC000001's video, after this RX is switched to OIP-A60E-DCE2AC000002, RX will still display the same position of OIP-A60E-DCE2AC000002'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 OIP-A60D-DCE2AC000104 and OIP-A60D-DCE2AC000103 to be switched to TX OIP-A60E-DCE2AC000003, RX OIP-A60D-DCE2AC000102 to TX OIP-A60E-DCE2AC000002, and RX OIP-A60D-DCE2AC000101 to TX OIP-A60E-DCE2AC000001:

Command:

```
matrix set OIP-A60E-DCE2AC000003 OIP-A60D-DCE2AC000104 OIP-A60D-DCE2AC000103,
OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000102, OIP-A60E-DCE2AC000001
OIP-A60D-DCE2AC000101
```

Response:

```
matrix set OIP-A60E-DCE2AC000003 OIP-A60D-DCE2AC000104 OIP-A60D-DCE2AC000103,
OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000102, OIP-A60E-DCE2AC000001
OIP-A60D-DCE2AC000101
```

Example2:

If you want RX OIP-A60D-DCE2AC000104 to stop decoding video:

Command:

```
matrix set NULL OIP-A60D-DCE2AC000104
```

Response:

```
matrix set NULL OIP-A60D-DCE2AC000104
```

2.2.2 matrix get

Command	`matrix get [OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102 ...]`

Response	`matrix information:`
	`OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104`
	`...`

Description	Obtains the source played by each receiver in the matrix. Omit receiver names to obtain the complete matrix, or specify one or more receiver names to obtain only those receivers. If TX is NULL, the receiver does not decode video. "NULL" is not case sensitive.

Example:

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

Command:

```
matrix get
```

Response:

```
matrix information:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
null OIP-A60D-DCE2AC000104
```

The response indicates that OIP-A60D-DCE2AC000101, OIP-A60D-DCE2AC000102, and OIP-A60D-DCE2AC000103 all playTX1, and that OIP-A60D-DCE2AC000104 does not decode video.

2.2.3 matrix video set

Command	`matrix video set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Response	`matrix video set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
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:

To route OIP-A60E-DCE2AC000001 to OIP-A60D-DCE2AC000101 and OIP-A60D-DCE2AC000102:

Command:

```
matrix video set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

Response:

```
matrix video set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

2.2.4 matrix video get

Command	`matrix video get OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102……`

Response	`matrix video information:`
	`OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104`
	`...`

Description	Obtain all or parts of RX video matrix information. Note: if the command doesn'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:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104
```

2.2.5 matrix audio set

Command	`matrix audio set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Response	`matrix audio set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Description	Set audio matrix link relationship between TX and RX. A certain or a few RXs are linked to a certain or some TXs. Each record of TX and its associated RX will be separated by a comma. If TX is null, it means audio matrix link relationship between the TX and RX has been removed.

Example:

If you want to route a TX's audio to a RX in matrix:

Command:

```
matrix audio set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101,
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104, OIP-A60E-DCE2AC000001
OIP-A60D-DCE2AC000103, OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
```

Response:

```
matrix audio set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101,
OIP-A60E-DCE2AC000001
OIP-A60D-DCE2AC000104, OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103,
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
```

2.2.6 matrix audio get

Command	`matrix audio get OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102……`

Response	`matrix audio information:`
	`OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104`
	`...`

Description	Obtain all or specified audio matrix information of RX(s). If any RX is not specified in this command, you will obtain audio matrix information of all RXs.

Example:

If you want to obtain all RX audio matrix information:

Command:

```
matrix audio get
```

Response:

```
matrix audio information:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104
```

2.2.7 matrix usb set

Command	`matrix usb set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Response	`matrix usb set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
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

Example:

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

Command:

```
matrix usb set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101, OIP-A60E-DCE2AC000001
 OIP-A60D-DCE2AC000104, OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103,
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
```

Response:

```
matrix usb set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101, OIP-A60E-DCE2AC000001

OIP-A60D-DCE2AC000104, OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103,
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
```

2.2.8 matrix usb get

Command	`matrix usb get OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102……`

Response	`matrix usb information:`
	`OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104`
	`...`

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

Example:

If you want to obtain all RX usb matrix information:

Command:

```
matrix usb get
```

Response:

```
matrix usb information:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104
```

2.2.9 matrix audio3 set

Command	`matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002, OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000003 OIP-A60E-DCE2AC000004, ...`
Response	`matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002, OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000003 OIP-A60E-DCE2AC000004, ...`
Description	Sets the ARC audio return matrix. Each record starts with an RX and is followed by one or more TX devices that listen to that RX. Separate records with commas. If RX is NULL, the existing binding for the listed TX devices is cleared. ALL_TX represents all TX devices.

Example 1:

If you want to route ARC audio from OIP-A60D-DCE2AC000101 to OIP-A60E-DCE2AC000001 and OIP-A60E-DCE2AC000002:

Command:

```
matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001
OIP-A60E-DCE2AC000002
```

Response:

```
matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001
OIP-A60E-DCE2AC000002
```

Example 2:

If you want to route ARC audio from OIP-A60D-DCE2AC000101 to OIP-A60E-DCE2AC000001 and OIP-A60E-DCE2AC000002, and from OIP-A60D-DCE2AC000102 to OIP-A60E-DCE2AC000003 and OIP-A60E-DCE2AC000004:

Command:

```
matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001
OIP-A60E-DCE2AC000002, OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000003
OIP-A60E-DCE2AC000004
```

Response:

```
matrix audio3 set OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001
OIP-A60E-DCE2AC000002, OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000003
OIP-A60E-DCE2AC000004
```

2.2.10 matrix audio3 get

Command	`matrix audio3 get OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002`

Response	`matrix audio3 information:`
	`OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001`
	`OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000002`
	`OIP-A60D-DCE2AC000102 OIP-A60E-DCE2AC000004`
	`...`

Description	Retrieves 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
OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000001
OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000002
```

2.2.11 matrix infrared set

Command	`matrix infrared set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
Response	`matrix infrared set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...`
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 OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

Response:

```
matrix infrared set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

2.2.12 matrix infrared get

Command	`matrix infrared get OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102`

Response	`matrix infrared information:`
	`OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103`
	`OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104`
	`...`

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:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104
```

2.2.13 matrix serial set

Command	matrix serial set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...
Response	matrix serial set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102,OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104,...
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 OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

Response:

```
matrix serial set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60D-DCE2AC000102
```

2.2.14 matrix serial get

Command	matrix serial get OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102
Response	matrix serial information: OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000104 ...
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:
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000102
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000103
OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000104
```

2.3 scene Commands

The following commands provide scene configuration and switching functions.

```
Note: Before using these commands the video wall and the layout called 'Office-
MeetingRoom' or 'scene1' with "Standard Video Wall" type must be created in advance
with the configurator.
```

2.3.1 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-MeetingRoomOffice-TrainingRoomOffice-TeaRoom
```

2.3.2 scene active

Command	`scene active <scene_name>`
Response	`scene active <scene_name> 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 active Office-MeetingRoom success
```

2.3.3 scene set

Command	`scene set <scene_name> posX posY <tx_name1>…`
Response	`scene <scene_name>'s source in [posX,posY] change to <tx_name1>`
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 (OIP-A60E-DCE2AC000001) to RX in scene Office-MeetingRoom of video wall:

Command:

```
scene set Office-MeetingRoom 1 2 OIP-A60E-DCE2AC000001
```

Response:

```
scene Office-MeetingRoom's source in [1 2] change to OIP-A60E-DCE2AC000001
```

2.3.4 scene change scenename txname

Command	scene change <scene_name> <txname>
Response	scene <scene_name>'s tx change to <tx_name1>
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 (OIP-A60E-DCE2AC000001) to all RX in scene1 of video wall:

Command:

```
scene change scene1 OIP-A60E-DCE2AC000001
```

Response:

```
scene scene1's tx change to OIP-A60E-DCE2AC000001
```

2.3.5 scene set bezelgap

Command	scene set <scene_name> bezelgap <video_wall_name> ow oh vw vh
Response	scene set <scene_name> bezelgap <video_wall_name> ow 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
```

2.3.6 scene set stretch

Command	scene set <scene_name> stretch <video_wall_name> type
Response	scene set <scene_name> stretch <video_wall_name> type success
Description	Sets the stretch mode for a specific video wall in a scene. sceneName specifies the target scene name. vw-name specifies the video wall name. type: Specifies the display mode. Supported values: fit: Scales the image proportionally to fit within the display area. May introduce black borders. stretch: Stretches the image to fill the display area without preserving aspect ratio. No borders appear. fill: Scales the image proportionally to fully cover the display area. May crop part of the image. Default value: stretch

Example:

To set the stretch mode of a video wall in a scene:

Command:

```
scene set Scene1 bezelgap vwtest5 fit
```

Response:

```
scene set Scene1 bezelgap vwtest5 fit success
```

2.3.7 scene connect scenename

Command	`scene connect <scene_name> <tx_name1> <tx_name2> … <tx_name_n>`
Response	`scene connect <scene_name> <tx_name1> <tx_name2> … <tx_name_n> success`
Description	Assigns sources to the corresponding RX of a scene in sequence when you want displays formed a video wall to show various sources. This action is operated only once and will not be saved in IP controller.

Example:

If you want to assign sources (OIP-A60E-DCE2AC000001, OIP-A60E-DCE2AC000002, OIP-A60E-DCE2AC000003, OIP-A60E-DCE2AC000004) to the corresponding RX of scene1 in sequence:

Command:

```
scene connect scene1 OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000003 OIP-A60E-DCE2AC000004
```

Response:

```
scene connect scene1 OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002
OIP-A60E-DCE2AC000003 OIP-A60E-DCE2AC000004 success
```

2.3.8 scene create

Command	`scene create <scene_json>`
Response	`scene create success`
Description	Creates scene configurations according to the specified JSON string.

Note:
scenesjsonstring: Specifies the scene configuration in standard JSON format.
The JSON string must not contain carriage returns or line breaks.
Multiple scene configurations can be included in one JSON string. The controller stores the configuration information of these scenes.

Example:

MAC-derived hostnames in this example are illustrative. Replace them with the actual hostnames accepted by the controller.

This example sends one complete scene from the source configuration. Serialize the JSON on one line.

Command:

```
scene create [{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"m":2,"n":2
,"name":"scene2-2x22","rxArray":[[{"aliasName":"visualm7","devName":"OIP-A60D-
DCE2AC000104","deviceType":"Receiver","group":[{"name":"group1","sequence":0}],"o
nline":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000005","txName":"OIP-A60E-
DCE2AC000002"},{"aliasName":"visualm12","devName":"OIP-A60D-
DCE2AC000103","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000006","txName":"OIP-
A60E-DCE2AC000002"}],[{"aliasName":"visualm8","devName":"OIP-A60D-
DCE2AC000102","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000007","txName":"OIP-
A60E-DCE2AC000002"},{"aliasName":"visualm13","devName":"OIP-A60D-
DCE2AC000101","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000101","txName":"OIP-
A60E-
DCE2AC000001"}]],"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}]}]
```

Response:

```
scene create success
```

See Scene Data for the complete JSON structure.

2.3.9 scene update

Command	`scene update <scene_json>`
Response	`scene update success`
Description	Updates the scene information described in scenesjsonstring. Note: scenesjsonstring: Specifies the scene configuration in standard JSON format. The JSON string must not contain carriage returns or line breaks. Multiple scene configurations can be included in one JSON string. The controller stores the configuration information of these scenes.

Example:

MAC-derived hostnames in this example are illustrative. Replace them with the actual hostnames accepted by the controller.

This example sends one complete scene from the source configuration. Serialize the JSON on one line.

Command:

```
scene update [{"group":[{"name":"group1","sequence":0}],"layoutseq":0,"m":2,"n":2
,"name":"scene2-2x22","rxArray":[[{"aliasName":"visualm7","devName":"OIP-A60D-
DCE2AC000104","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000005","txName":"OIP-
A60E-DCE2AC000002"},{"aliasName":"visualm12","devName":"OIP-A60D-
DCE2AC000103","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000006","txName":"OIP-
A60E-DCE2AC000002"}],[{"aliasName":"visualm8","devName":"OIP-A60D-
DCE2AC000102","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000007","txName":"OIP-
A60E-DCE2AC000002"},{"aliasName":"visualm13","devName":"OIP-A60D-
DCE2AC000101","deviceType":"Receiver","group":[{"name":"undefined","sequence":0}]
,"online":true,"sequence":0,"trueName":"OIP-A60D-DCE2AC000101","txName":"OIP-
A60E-
DCE2AC000001"}]],"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}]}]
```

Response:

```
scene update success
```

See Scene Data for the complete JSON structure.

2.3.10 scene modify name

Command	`scene modify name scenename_old1 scenename_new1 scenename_old2 scenename_new2 ...`
Response	`scene modify name scenename_old1 scenename_new1 scenename_old2 scenename_new2...success`
Description	Modifies scene names.

Example:

To change the scene name “vw-scene1” to “vw-scene2”:

Command:

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

Response:

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

2.3.11 scene remove

Command	`scene remove scenename1 scenename2 ...`
Response	`scene remove scenename1 scenename2 ... success`
Description	Removes scenes.

Example:

To remove the scene “vw-scene1”:

Command:

```
scene remove vw-scene1
```

Response:

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

2.3.12 scene group update

Command	`scene group update scenejsonstring`
Response	`scene group or sort update success`
Description	Updates scene group assignments and scene sort order using scenejsonstring. Note: • ungrouped is the default system group and has sequence number 1. User-defined groups start at sequence number 2.

Example:

To assign a scene to group1 and update its sort order:

Command:

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

Response:

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

2.4 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.

2.5 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.

2.5.1 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 OIP-A60E-DCE2AC000001 just got online.

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify endpoint + OIP-A60E-DCE2AC000001
```

2.5.2 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:

OIP-A60D-DCE2AC000101'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"):

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify serialinfo OIP-A60D-DCE2AC000101 hex 56:
68 65 6C 11 6C 6F 11 22 33 44 00 55 66 77 99 AA CC DD FF
```

Example2:

OIP-A60D-DCE2AC000102's serial port receives five characters "12345":

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify serialinfo OIP-A60D-DCE2AC000102 ascii 5:
12345
```

2.5.3 notify device settings_changed

Command	`notify device settings_changed hostname1 hostname2...`
Description	Proactively notifies connected clients that the settings of one or more devices have changed. Note: • This notification is available when config set system notify_enable is set to on.

Example:

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify device settings_changed OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

2.5.4 notify device status_changed

Command	`notify device status_changed hostname1 hostname2...`
Description	Proactively notifies connected clients that the status of one or more devices has changed. Note: • This notification is available when config set system notify_enable is set to on.

Example:

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify device status_changed OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000001
```

2.5.5 notify video

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:

```
OIP-A60E-DCE2AC000001 got the video lost from the system, then the IP controller will offer the notification like under:
```

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify video lost OIP-A60E-DCE2AC000001
```

2.5.6 notify sink

Command	`notify link {lost|found} hostname`
Description	Proactively notifies a connected third-party control device when the HDMI output signal on a decoder is lost or restored. Note: hostname is the name of the decoder reporting the sink-link status.

Example:

```
The controller reports that the HDMI output signal of OIP-A60D-DCE2AC000101 is lost:
```

Response format illustration. Content enclosed in angle brackets is a placeholder, not an actual returned value.

```
notify link lost OIP-A60D-DCE2AC000101
```

2.5.7 notify update

Command	`notify update DEV status`
Description	The controller proactively notifies a third-party control server of a device's upgrade status. The following status types are supported: 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. error: indicates an upgrade error occurred (e.g., network disconnection). The device may still be upgrading normally. timeout: indicates the upgrade has timed out due to no response from the device for an extended period; the device may still be upgrading normally. Note: "notify endpoint" only indicates a failure to connect to the device via Telnet or SSH. It does NOT indicate a failure of the upgrade process itself, which may use HTTP or other transport methods.

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

Example 1:

```
notify update OIP-A60D-DCE2AC000101 success
```

Example 2:

```
notify update OIP-A60D-DCE2AC000101 processing 56
```

2.5.8 notify ntpserver is {reachable | unreachable}

Command	`notify ntpserver is {reachable | unreachable}`
Description	Proactively notifies a connected third-party control device when the configured NTP server becomes reachable or unreachable.

Example:

The controller reports that the configured NTP server is reachable:

```
notify ntpserver is reachable
```

2.6 Infrared Command

2.6.1 Infrared Command

Command	`infrared "IRDATA" <hostname>`
Response	`infrared "IRDATA" <hostname>`
Description	Sends the specified infrared data “IRDATA” to the target device. The infrared data must be enclosed in double quotation marks ("").

Example:

To send IR data in Pronto format to a specific device:

Command:

```
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
```

Note

When this command is sent to the controller, it is internally translated into an irs command and forwarded to the specified device.

2.7 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
```

3 Concurrent Commands

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

Commands
matrix set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC000101 OIP-A60D-DCE2AC000102, OIP-A60E-DCE2AC000002 OIP-A60D-DCE2AC000103 OIP-A60D-DCE2AC000104, ...
scene active scenename
scene connect scenename OIP-A60E-DCE2AC000001 OIP-A60E-DCE2AC000002 … txnm

Example:

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

Command:

```
matrix set OIP-A60E-DCE2AC000001 OIP-A60D-DCE2AC00010B OIP-A60D-DCE2AC00010C
OIP-A60D-DCE2AC00010D; scene connect scene1 OIP-A60E-DCE2AC000001
OIP-A60E-DCE2AC000002 OIP-A60E-DCE2AC000003 OIP-A60E-DCE2AC000004
```

4 Data Structures

This chapter describes shared data structures. Refer to each API for its request syntax and applicable conditions. JSON is expanded here for readability; requests are transmitted on one line.

4.1 Scene Data

Shared scene structure for config get scenejsonstring, scene create and scene update. Refer to each API for its applicability, conditions and operation. Serialize requests on one line. The structure below is a representative scene; it does not define undocumented required fields or defaults.

Field index — names and example types below are derived from the retained source illustration. Requiredness, defaults and undocumented constraints remain TBC; this index does not establish new protocol guarantees.

Field	Example type
group	array
layoutseq	number
m	number
n	number
name	string
rxArray	array
sceneAutoApply	boolean
sequence	number
txListArray	array
vwConfigList	array
rxArray[][].devName	string; present in the source create/update examples
rxArray[][].rxstatus	Type TBC; present in the source query illustration

```
[
    {
        "group": [
            {
                "name": "group1",
                "sequence": 0
```

```
    }
],
"layoutseq": 0,
"m": 2,
"n": 2,
"name": "scene2-2x22",
"rxArray": [
    [
        {
            "aliasName": "visualm7",
            "devName": "OIP-A60D-DCE2AC000104",
            "deviceType": "Receiver",
            "group": [
                {
                    "name": "group1",
                    "sequence": 0
                }
            ],
            "online": true,
            "sequence": 0,
            "trueName": "OIP-A60D-DCE2AC000005",
            "txName": "OIP-A60E-DCE2AC000002"
        },
        {
            "aliasName": "visualm12",
            "devName": "OIP-A60D-DCE2AC000103",
            "deviceType": "Receiver",
            "group": [
                {
                    "name": "undefined",
                    "sequence": 0
                }
            ],
            "online": true,
            "sequence": 0,
            "trueName": "OIP-A60D-DCE2AC000006",
            "txName": "OIP-A60E-DCE2AC000002"
        }
    ],
    [
        {
            "aliasName": "visualm8",
            "devName": "OIP-A60D-DCE2AC000102",
            "deviceType": "Receiver",
            "group": [
                {
                    "name": "undefined",
                    "sequence": 0
                }
            ],
            "online": true,
            "sequence": 0,
            "trueName": "OIP-A60D-DCE2AC000007",
            "txName": "OIP-A60E-DCE2AC000002"
        },
        {
            "aliasName": "visualm13",
            "devName": "OIP-A60D-DCE2AC000101",
            "deviceType": "Receiver",
            "group": [
                {
                    "name": "undefined",
                    "sequence": 0
                }
            ],
            "online": true,
```

```
                    "sequence": 0,
                    "trueName": "OIP-A60D-DCE2AC000101",
                    "txName": "OIP-A60E-DCE2AC000001"
                }
            ]
        ],
        "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
            }
        ]
    }
]
```


4.2 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
name	Device name. Read only. Format is "Device type-MAC address" such as the illustrative hostname OIP-A60E-DCE2AC000001
version	Device software version. Read only. Format is v#.#.# such as v2.5.6
mac	Device MAC address. Read only.
ip_mode	IP address obtain method. autoip: AutoIP static: Static IP dhcp: DHCP
ip4addr	IPv4 address.When ip_mode is static, IPv4 address can be changed.
netmask	IPv4 subnet mask. When ip_mode is static, IPv4 subnet mask can be changed.

Parameters	Description
gateway	IPv4 gateway.When ip_mode is static, IPv4 gateway can be changed.

4.2.1 Decoder — OIP-A60D

Applies to	OIP-A60D

Requires: Decoder role

Requires: Decoder role

Parameter	Description	Example	Applies to
output_force_resolution	Specifies the forced output resolution. The value includes auto, pass-through, 2160p60, 2160p50, 2160p30, 2160p25, 2160p24, 1080p60, 1080p50, 1080p30, 1080p25, 1080p24, 1080i60, 1080i50, 720p60, 720p50, 720p30, 720p25, and 720p24.	output_force_resolution=1080p60	OIP-A60D
hdcp_out_mode	Sets the HDCP output mode. Available values: follow_sink, follow_source, always1.4, and always2.2.	hdcp_out_mode=always1.4	OIP-A60D
turn_off_screen	Determine whether to turn off screen when there’s no source. y/n	turn_off_screen=y	OIP-A60D
src_unavailable_timeout	The timeout that detects signal lost. Range: [1,60] -1: Timeout doesn’t occur	src_unavailable_timeout=50	OIP-A60D
video_stretch_type	stretch_out/fit_in	video_stretch_type=fit_in	OIP-A60D
video_rotate	Video rotation angle: 0°/180°/270°	video_rotate=180	OIP-A60D
video_gen_lock	Video genlock setting, [0-FFFFFFFF]	video_gen_lock=22334455	OIP-A60D
audio_addon	Audio addon selection. none (Default) dante When audio_addon = dante, Dante feature is enabled. When audio_addon = None, Dante feature is disabled.	audio_addon=dante	OIP-A60D
dante_op_mode	Dante AV-A operation mode. AV-A (Default) audio-only Note: Settings take effect after device reboots.	dante_op_mode=AV-A	OIP-A60D
dante_trial_mode	Enable Dante trial mode without activation (one hour for free testing). y n (Default)	dante_trial_mode=y	OIP-A60D
audio_bridge_en	Enable bridge between two audio networks y n (Default)	audio_bridge_en=y	OIP-A60D
audio_out_src_sel	Source selection for I2S output. native (Default) addon	audio_out_src_sel=addon	OIP-A60D

Parameter	Description	Example	Applies to
audio_gen_lock	Audio genlock setting, [0-FFFFFFFF]	audio_gen_lock=22334455	OIP-A60D
enc_fps	[0,60]	enc_fps=60	OIP-A60D
rs232_baudrate	115200-8n1	rs232_baudrate=115200-8n1	OIP-A60D
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through	OIP-A60D
km_over_ip_enable	true/false	km_over_ip_enable=true	OIP-A60D
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true	OIP-A60D
audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable=true	OIP-A60D
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true	OIP-A60D
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false	OIP-A60D
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false	OIP-A60D
sinkpower.cec_mode	To enable CEC control on sinkpower: y/n	sinkpower.cec_mode=y	OIP-A60D
sinkpower.cec.onetouchplay	The CEC command value of OneTouchPlay; if there’re multiple CEC commands, use comma " ," to connect them.	sinkpower.cec.onetouchplay="4004"	OIP-A60D
sinkpower.cec.standby	The CEC command value of Standby; if there’re multiple CEC commands, use comma " ," to connect them.	sinkpower.cec.standby="ff36"	OIP-A60D
sinkpower.rs232_mode	To enable RS232 control on sinkpower: y/n	sinkpower.rs232_mode=y	OIP-A60D
sinkpower.rs232.mode	The format of the RS232 commands: hex/ascii	sinkpower.rs232.mode=ascii	OIP-A60D
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	OIP-A60D
sinkpower.rs232.onetouchplay	RS232 command that used to control OneTouchPlay.	sinkpower.rs232.onetouchplay="Power on"	OIP-A60D
sinkpower.rs232.standby	RS232 command that used to control StandBy.	sinkpower.rs232.standby="Power off"	OIP-A60D
sinkpower.ir_mode	To enable IR control on sinkpower: y/n	sinkpower.ir_mode=y	OIP-A60D
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"	OIP-A60D

Parameter	Description	Example	Applies to
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="Power off"	OIP-A60D
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} y: enable n: disable	ieee8021x.enable=y	OIP-A60D
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls	OIP-A60D
ieee8021x.mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user	OIP-A60D
ieee8021x.mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd	OIP-A60D
ieee8021x.tlsuser	TIS user name	ieee8021x.tlsuser=user	OIP-A60D
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikeypwd	OIP-A60D
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y	OIP-A60D
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	OIP-A60D
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	OIP-A60D
ui_show_text	Turns on/off OSD display, y/n.	ui_show_text=y	OIP-A60D
audio_analog_out_mute	Enables or disables analog audio output mute. The value includes y and n.	audio_analog_out_mute=y	OIP-A60D
lan_mode	Specifies the LAN operating mode. The value includes switched, independent, and bridged.	lan_mode=switched	OIP-A60D

4.2.2 Encoder — OIP-A60E

Applies to	OIP-A60E

Requires: Encoder role

Requires: Encoder role

Parameter	Description	Example	Applies to
audio_input_type	Available audio input channels: auto, hdmi, analog	Read only.	OIP-A60E
audio_analog_out_vol	Range of the analog audio output volume: [0,100]	audio_analog_out_vol=90	OIP-A60E
audio_direction	Settings of audio direction: in, out.	audio_direction=in	OIP-A60E
preview_enable	Enable/disable preview: y/n	preview_enable=y	OIP-A60E
bit_rate	Settings of bit rate: 50M–811M	Bit_rate=100M	OIP-A60E
hdcp_enable	y: Enable HDCP support n: Disable HDCP support	hdcp_enable=y	OIP-A60E

Parameter	Description	Example	Applies to
audio_addon	none (default) dante	audio_addon=dante	OIP-A60E
dante_op_mode	Dante AV-A operation mode: AV-A (default) dante Note: Settings take effect after device reboots.	dante_op_mode=AV-A	OIP-A60E
dante_trial_mode	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	OIP-A60E
audio_bridge_en	Enable/disable bridge between two audio networks. y n (Default)	audio_bridge_en=y	OIP-A60E
audio_out_src_sel	Source selection for I2S output. native (Default) addon	audio_out_src_sel=addon	OIP-A60E
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	OIP-A60E
rs232_working_mode	RS232 work mode: Feedback/Pass-through	rs232_working_mode=feed_back rs232_working_mode=pass_through	OIP-A60E
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	OIP-A60E
video_over_ip_enable	To enable video over IP: true/false	video_over_ip_enable=true	OIP-A60E
audio_over_ip_enable	To enable audio over IP: true/false	audio_over_ip_enable=true	OIP-A60E
usb_over_ip_enable	To enable usb over IP: true/false	usb_over_ip_enable=true	OIP-A60E
km_over_ip_enable	true/false	km_over_ip_enable=false	OIP-A60E
serial_over_ip_enable	To enable serial over IP: true/false	serial_over_ip_enable=false	OIP-A60E
ir_over_ip_enable	To enable IR over IP: true/false	ir_over_ip_enable=false	OIP-A60E
edid	Reset EDID.	edid=reset	OIP-A60E
edid_hdmi	Reset HDMI EDID	edid_hdmi= reset	OIP-A60E
edid_usbc	Reset USBC EDID	edid_usbc=reset	OIP-A60E

Parameter	Description	Example	Applies to
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"	OIP-A60E
ieee8021x.enable	To enable or disable IEEE 8021.x settings. value: {y|n} y: enable n: disable	ieee8021x.enable=y	OIP-A60E
ieee8021x.mode	mode: mschapv2/tls	ieee8021x.mode=tls	OIP-A60E
ieee8021x. mschapv2user	mschapv2 user name	ieee8021x.mschapv2user=user	OIP-A60E
ieee8021x. mschapv2pwd	mschapv2 password	ieee8021x.mschapv2pwd=pwd	OIP-A60E
ieee8021x.tlsuser	TIS user name	ieee8021x.tlsuser=user	OIP-A60E
ieee8021x.tlsprikeypwd	private key	ieee8021x.tlsprikeypwd=prikeypwd	OIP-A60E
ieee8021x.camode	Determine whether CA certificate is authenticated: y/n	ieee8021x.camode=y	OIP-A60E
audio_analog_out_mute	Enables or disables analog audio output mute. The value includes y and n.	audio_analog_out_mute=y	OIP-A60E