API COMMAND SET

WST-PMX006

HDMI 4K60Hz 8x8 Seamless Matrix with Audio Extraction

V1.0.0| July 15, 2026

RS232 Default Setting

Parameters	Value
Baud Rate	9600 bps
Data bits	8 bits
Parity	None
Stop bits	1 bit
Flow control	None

Telnet Session Setup

Before sending API commands to the device's LAN port, telnet session between the device and the PC should be established.

To establish a telnet session, send the following command:

telnet ip (port)

- *ip*: The device's IP address.
- *port:* The protocol port number, this is non-required on some telnet client tools or platforms. By default, the telnet port number is 23 and telnet-TLS port number is 24.

For example, if the device's IP address is 192.168.10.254, the command for establishing a telnet session shall be:

telnet 192.168.10.254 23

Command Description

Take Command *SET SW in out<CR><LF>* as an example:

1. *[SET SW]* denotes command key words, case insensitive.
2. *[in out]* denotes parameters, case insensitive; incorrect parameters will not be recognized.
3. *<CR><LF>* denotes a carriage return or a line feed; all commands must be ended up with a carriage return or a line feed.

No.	Description	Command	Example
		Video Switch	
1	Switch one input to one output	**Command:** SET SW *in out*<CR><LF> **Return:** SW *in out*<CR><LF> **Parameter:** *in* = {in1, in2…in8, in0}; *out* = {out1, out2…out8}; **Description:** SW is short for Switch Switch one input to one output.	**Command:** SET SW in1 out2<CR><LF> **Return:** SW in1 out2<CR><LF> **Description:** Switch input 1 to hdmi output 2.
2	Switch one input to all outputs	**Command:** SET SW *in all*<CR><LF> **Return:** SW *in all* <CR><LF> **Parameter:** *in* = { in1, in2… in8, in0}; *all* = {all}; **Description:** SW is short for Switch Switch one input to all outputs.	**Command:** SET SW in1 *all* <CR><LF> **Return:** SW in1 all<CR><LF> **Description:** Switch input1 to all output sinks.
3	Get which input is switched to one indicated output	**Command:** GET MP *out*<CR><LF> **Return:** Mp *in out*<CR><LF> **Parameter:** *in* = {in1, in2…in8, in0}; *out* = {out1, out2…out8}; **Description:** MP is short for mapping Get which input is switched to one indicated output.	**Command:** GET MP *out1*<CR><LF> **Return:** MP *hdmiin2 out1*<CR><LF> **Description:** Get which input is switched to out1, the result is hdminin2.
4	Get correspondence between all inputs and outputs	**Command:** GET MP *all*<CR><LF> **Return:** MP *in out*<CR><LF> MP *in out*<CR><LF> ..... **Parameter:** in = {in1, in2…in8，in0}; out = {out1, out2…out8}; *all* = {all}; **Description:** MP is short for mapping Get correspondence between all inputs and outputs.	**Command:** GET MP *all* <CR><LF> **Return:** MP *in1 out1*<CR>... MP *in2 out2*<CR><LF> …... **Description:** Get correspondence between all inputs and outputs.

No.	Description	Command	Example
5	Set one output AV MUTE to on/off	**Command:** SET AV_MUTE *out prm*<CR><LF> **Return:** AV_MUTE *out prm*<CR><LF> **Parameter:** *out* = {out1, out2, out3, out4…out12}; *prm* = {on, off}// on means mute; off means unmute **Description:** Mute/unmute the output audio and video.	**Command:** SET AV_MUTE *out1 on* <CR><LF> **Return:** AV_MUTE *out1 on* <CR><LF> **Description:** Mute output1.
6	Get output AV mute status	**Command:** GET AV_MUTE *out*<CR><LF> **Return:** AV_MUTE out *prm*<CR><LF> **Parameter:** *out* = {out1, out2, out3, out4…out12}; *prm* = {on, off}// on means mute; off means unmute **Description:** Get output AV mute status.	**Command:** GET AV_MUTE *out1* <CR><LF> **Return:** AV_MUTE out1 *on* <CR><LF> **Description:** Get AV mute status of out1,and the status is mute.
Audio Switch			
1	Set the mode of audio switching	**Command:** SET AUDIOSW_M *prm*<CR><LF> **Return:** AUDIOSW_M *prm*<CR><LF> **Parameter:** *prm* = {independent, follow}; **Description:** Set the mode of audio switching.	**Command:** SET AUDIOSW_M *independent*<CR><LF> **Return:** AUDIOSW_M *independent*<CR><LF> **Description:** Set the audio switching mode to independent mode.
2	Get audio switching mode	**Command:** GET AUDIOSW_M<CR><LF> **Return:** AUDIOSW_M *prm*<CR><LF> **Parameter:** prm = {independent, follow}; **Description:** Get audio switching mode.	**Command:** GET AUDIOSW_M<CR><LF> **Return:** AUDIOSW_M independent<CR><LF> **Description:** Get audio switching mode.
3	Switch one audio source to one audio output	**Command:** SET AUDIOSW *in out*<CR><LF> **Return:** AUDIOSW *in out*<CR><LF> **Parameter:** in = {IN1~IN8}; *out* = {LINEOUT1~LINEOUT4, all}; **Description:** Audio channel switching supported only in standalone mode.	**Command:** SET AUDIOSW *IN1 LINEOUT1*<CR><LF> **Return:** AUDIOSW IN1 *LINEOUT1*<CR><LF> **Description:** In independent mode, switch the IN1 audio source to LINEOUT1.

No.	Description	Command	Example
4	Get which audio input is mapping to the indicate Output	**Command:** GET AUDIOMP *out*<CR><LF> **Return:** AUDIOMP *in out*<CR><LF> **Parameter:** in = {IN1~IN8}; out = {LINEOUT1~LINEOUT4, all}; **Description:** Get which audio input is mapping to the indicate Output.	**Command:** GET AUDIOMP *LINEOUT1*<CR><LF> **Return:** AUDIOMP IN1 LINEOUT1<CR><LF> **Description:** In independent mode, the audio source for LINEOUT1 is IN1
5	Volume gain adjustment	**Command:** SET VOLGAIN_DATA *out prm*<CR><LF> **Return:** VOLGAIN_DATA *aout prm*<CR><LF> **Parameter:** *out = {lineout1, lineout2,…lineout4};* *prm* = {-100 ~ 0} **Description:** VOLGAIN is volgain.	**Command:** SET VOLGAIN_DATA *lineout1 -50*<CR><LF> **Return:** VOLGAIN_DATA *lineout1*<CR><LF> **Description:** Set volume gain of lineout1 to -50.
6	Get output volume gain	**Command:** GET VOLGAIN_DATA *out*<CR><LF> **Return:** VOLGAIN_DATA *out prm*<CR><LF> **Parameter:** *out = {lineout1, lineout2,…lineout4};* *prm* = {-100~0} **Description:** VOLGAIN is volgain	**Command:** GET VOLGAIN_DATA *lineout1*<CR><LF> **Return:** VOLGAIN_DATA *lineout1 -50* <CR><LF> **Description:** Get volume gain ,the gain is -50
7	Mute/unmute output audio	**Command:** SET AUDIO_MUTE *out prm*<CR><LF> **Return:** AUDIO_MUTE *out prm*<CR><LF> **Parameter:** *aout* = {lineout1, lineout2,…lineout4}; *prm* = {on, off}// on means mute; off means unmute **Description:** MUTE is mute.	**Command:** SET AUDIO_MUTE *lineout1 on* <CR><LF> **Return:** AUDIO_MUTE *lineout1 on* <CR><LF> **Description:** Mute lineout 1.
8	Get if the output audio is in mute status	**Command:** GET AUDIO_MUTE *out*<CR><LF> **Return:** AUDIO_MUTE out *prm*<CR><LF> **Parameter:** out = {lineout1, lineout2,…lineout4}; *prm* = {on, off}// on means mute; off means unmute **Description:** MUTE is mute.	**Command:** GET AUDIO_MUTE *lineout1* <CR><LF> **Return:** AUDIO_MUTE lineout1 *on* <CR><LF> **Description:** Get mute status of lineout1,and the status is mute.

No.	Description	Command	Example
		CEC	
1	Set CEC POWER ON/OFF	**Command:** SET CEC_PWR *out prm*<CR><LF> **Return:** CEC_PWR *out prm*<CR><LF> **Parameter:** *prm* = {on, off} *out* = {out1, out2…out8, all}; **Description:** Set CEC POWER ON/OFF.	**Command:** SET CEC_PWR out1 *on*<CR><LF> **Return:** CEC_PWR out1 *on*<CR><LF> **Description:** Set sink hdmi output 1 to power on via CEC.
2	Set CEC AUTO POWER ON/OFF	**Command:** SET AUTOCEC_FN *out prm*<CR><LF> **Return:** AUTOCEC_FN *out prm*<CR><LF> **Parameter:** *prm* = {on, off} *out* = {out1, out2…out8, all}; **Description:** Set sink auto power Function to ON or OFF.	**Command:** SET AUTOCEC_FN *out1 on*<CR><LF> **Return:** AUTOCEC_FN out1 *on*<CR><LF> **Description:** Set sink hdmi output 1 auto power to ON.
3	Get CEC AUTO POWER ON/OFF Status	**Command:** GET AUTOCEC_FN *out*<CR><LF> **Return:** AUTOCEC_FN *out prm*<CR><LF> **Parameter:** *prm* = {on,off} *out* = {out1,out2…out8}; **Description:** Get Sink auto power Function ON or OFF Status.	**Command:** GET AUTOCEC_FN out1<CR><LF> **Return:** AUTOCEC_FN *on* **Description:** Get Sink auto power status, and the status is ON.
4	Set CEC POWER Delay Time	**Command:** SET AUTOCEC_D *out prm*<CR><LF> **Return:** AUTOCEC_D *out prm*<CR><LF> **Parameter:** *out* = {out1, out2…out8, all}; *prm* = {1, 2, 3…,}// according to the actual time counter,1 means 1 minute ,2 means 2 minutes，Default wait time is 2 minutes, Max wait time is 30 minutes. 0 means when no active signal, the unit auto power off immediately. **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing.	**Command:** SET AUTOCEC_D *out1 2*<CR><LF> **Return:** AUTOCEC_D out1 *2*<CR><LF> **Description:** When there's no active signal to hdmi1, 2 minutes later, the unit will auto power off.

No.	Description	Command	Example
5	Get CEC POWER Delay Time Status	**Command:** GET AUTOCEC_D *out* <CR><LF> **Return:** AUTOCEC_D *out prm*<CR><LF> **Parameter:** *out* = {out1, out2…out8, all}; *prm* = {1,2,3…,}// according to the actual time counter,1 means 1 minute, 2 means 2 minutes，Default wait time is 2 minutes, Max wait time is 30 minutes. 0 means when no active signal, the unit auto power off immediately. **Description:** AUTOCEC_D is short for CEC auto Power Delay Timing.	**Command:** GET AUTOCEC_D out1 <CR><LF> **Return:** AUTOCEC_D *out1 2* <CR><LF> **Description:** Get hdmi1 auto power delay time, the result is 2 minutes.
6	Set CEC command for defining Poweron/Poweroff	**Command:** SET CECCMD_EDIT *out prm1 prm2*<CR><LF> **Return:** CECCMD_EDIT *out prm1 prm2<CR><LF>* **Parameter:** *out* = {out1, out2…out8,all}; prm1 = {pwron, pwroff}, // *prm2* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Set CEC command for defining Poweron/Poweroff. If relevant commands are required, please refer to CEC specification for details.	**Command:** SET CECCMD_EDIT out1 pwron 40 04<CR><LF> **Return:** CECCMD_EDIT out1 pwron 40 04<CR><LF> **Description:** Set CEC command for defining Poweron out1.
7	Get CEC command for defining Poweron/Poweroff	**Command:** GET CECCMD_EDIT *out prm1*<CR><LF> **Return:** CECCMD_EDIT *out prm1 prm2<CR><LF>* **Parameter:** *out* = {out1, out2…out8, all}; prm1 = {pwron, pwroff}, // *prm2* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Get CEC command for defining Poweron/Poweroff.	**Command:** GET CECCMD_EDIT out1 pwron<CR><LF> **Return:** CECCMD_EDIT out1 pwron 40 04<CR><LF>
8	Send CEC command	**Command:** SET CEC_CMD *out prm*<CR><LF> **Return:** CEC_CMD *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out8,all}; *prm* = {...},// many hexadecimal numbers, the total is less than 16. Space separated **Description:** Send CEC command. If relevant commands are required, please refer to CEC specification for details.	**Command:** SET CEC_CMD out1 40 04<CR><LF> **Return:** CEC_CMD out1 40 04<CR><LF> **Description:** Send CEC command.

No.	Description	Command	Example
		HDCP	
1	Set Input HDCP support to ON/OFF	**Command:** SET HDCP_S in prm<CR><LF> **Return:** HDCP_S *in prm*<CR><LF> **Parameter:** *prm* = {on, off} *in* = {in1…in8} **Description:** HDCP_S will control source hdcp support on or off.	**Command:** SET HDCP_S in2 *on*<CR><LF> **Return:** HDCP_S in2 *on*<CR><LF> **Description:** Set hdmi input 2 hdcp support to on.
2	Get Input HDCP support ON/OFF Status	**Command:** GET HDCP_S *in* <CR><LF> **Return:** HDCP_S *in prm*<CR><LF> **Parameter:** *prm* = {on, off} *in* = {in1…in8} **Description:** HDCP_S is short for HDCP support	**Command:** GET HDCP_S in2<CR><LF> **Return:** HDCP_S *in2 on*<CR><LF> **Description:** Get hdmi2 hdcp support on or off status, and the result is on.
3	Set Output HDCP	**Command:** SET HDCP *out prm*<CR><LF> **Return:** HDCP *out prm*<CR><LF> **Parameter:** *out* = {out1, out2…out8, all};// all includes hdmiout1-hdmiout8 *prm* = { follow, hdcp1.x}// follow means source use hdcp 1.4 ,then output also use hdcp 1.4. the same with hdcp 2.2 and off. **Description:** Set Output HDCP.	**Command:** SET HDCP *out1 follow*<CR><LF> **Return:** HDCP *out1 follow*<CR><LF> **Description:** Set output1 hdcp to follow mode.
4	Get Output HDCP	**Command:** GET HDCP *out*<CR><LF> **Return:** HDCP *out prm*<CR><LF> **Parameter:** *out* = {out1,out2…out8,all};// all includes hdmiout1-hdmiout8 *prm* = {follow, hdcp1.x}// follow means source use hdcp 1.4, then output also use hdcp 1.4. the same with hdcp 2.2 and off. **Description:** Get Output HDCP.	**Command:** GET HDCP *out1*<CR><LF> **Return:** HDCP *out1 follow*<CR><LF> **Description:** Get output1 hdcp, the result is follow mode.

No.	Description	Command	Example
		EDID	
1	Set Input EDID	**Command:** SET EDID *in prm*<CR><LF> **Return:** EDID *in prm*<CR><LF> **Parameter:** *in* = {in1, in2...in8}; prm = {1 ~30} 1: Copy form hdmi output 1 2: Copy form hdmi output 2 ... 8: Copy form hdmi output 8 9: Fixed 3840x2160@60 PCM Audio with HDR; 10: Fixed 3840x2160@60 PCM Audio with SDR; 11: Fixed 3840x2160@30 PCM Audio with HDR; 12: Fixed 3840x2160@30 PCM Audio with SDR; 13: Fixed 1920x1080@60 PCM Audio with HDR; 14: Fixed 1920x1080@60 PCM Audio with SDR; Special edid 15: Fixed 8Kx1K, 5Kx1K@60Hz PCM Audio 16: Fixed 3840x1080@60Hz PCM Audio 17: Fixed 3440x1440@60 PCM Audio 18: Fixed 2560x1600@60 PCM Audio 19: Fixed 2560x1536@60 PCM Audio 20: Fixed 2560x1440@60 PCM Audio 21: Fixed 2560x1080@60 PCM Audio 22: Fixed 1920x2160@60 PCM Audio 23: Fixed 1680x1050@60 PCM Audio 24: Fixed 1600x1200@60 PCM Audio 25. Fixed 5K Superwide PCM Audio with HDR 26. Fixed 5K Ultrawide PCM Audio with HDR **Description:** Set Input EDID.	**Command:** SET EDID *in1 10*<CR><LF> **Return:** EDID *in1 10*<CR><LF> **Description:** Set in1 EDID to Fixed 3840x2160@60 PCM Audio with SDR

No.	Description	Command	Example
2	Get All Input EDID status	**Command:** GET EDID *all* <CR><LF> **Return:** EDID *in prm*<CR> EDID *in prm*<CR> EDID *in prm*<CR><LF> ..... **Parameter:** in = {in1, in2...in8}; prm = {1 ~30} 1: Copy form hdmi output 1 2: Copy form hdmi output 2 ... 8: Copy form hdmi output 8 9: Fixed 4K60 2.0CH PCM Audio with HDR; 10: Fixed 4K60 2.0CH PCM Audio with SDR; 11: Fixed 4K30 2.0CH PCM Audio with HDR; 12: Fixed 4K30 2.0CH PCM Audio with SDR; 13: Fixed 1080p@60Hz 2.0CH PCM Audio with HDR; 14: Fixed 1080p@60Hz 2.0CH PCM Audio with SDR; Special edid 15: Fixed 8Kx1K, 5Kx1K@60Hz PCM Audio 16: Fixed 3840x1080@60Hz PCM Audio 17: Fixed 3440x1440@60 PCM Audio 18: Fixed 2560x1600@60 PCM Audio 19: Fixed 2560x1536@60 PCM Audio 20: Fixed 2560x1440@60 PCM Audio 21: Fixed 2560x1080@60 PCM Audio 22: Fixed 1920x2160@60 PCM Audio 23: Fixed 1680x1050@60 PCM Audio 24: Fixed 1600x1200@60 PCM Audio 25. Fixed 5K Superwide PCM Audio with HDR 26. Fixed 5K Ultrawide PCM Audio with HDR **Description:** Get all input EDID Status.	**Command:** GET EDID *all* <CR><LF> **Return:** EDID in1 *01*<CR> EDID in2 *02*<CR> EDID in3 *03*<CR><LF> …… **Description:** Get all input EDID Status.

No.	Description	Command	Example
3	Get one input EDID Status	**Command:** GET EDID *in* <CR><LF> **Return:** EDID *in prm*<CR><LF> **Parameter:** 1: Copy form hdmi output 1 2: Copy form hdmi output 2 ... 8: Copy form hdmi output 8 9: Fixed 4K60 2.0CH PCM Audio with HDR; 10: Fixed 4K60 2.0CH PCM Audio with SDR; 11: Fixed 4K30 2.0CH PCM Audio with HDR; 12: Fixed 4K30 2.0CH PCM Audio with SDR; 13: Fixed 1080p@60Hz 2.0CH PCM Audio with HDR; 14: Fixed 1080p@60Hz 2.0CH PCM Audio with SDR; Special edid 15: Fixed 8Kx1K, 5Kx1K@60Hz PCM Audio 16: Fixed 3840x1080@60Hz PCM Audio 17: Fixed 3440x1440@60 PCM Audio 18: Fixed 2560x1600@60 PCM Audio 19: Fixed 2560x1536@60 PCM Audio 20: Fixed 2560x1440@60 PCM Audio 21: Fixed 2560x1080@60 PCM Audio 22: Fixed 1920x2160@60 PCM Audio 23: Fixed 1680x1050@60 PCM Audio 24: Fixed 1600x1200@60 PCM Audio 25. Fixed 5K Superwide PCM Audio with HDR 26. Fixed 5K Ultrawide PCM Audio with HDR **Description:** Get one input EDID Status.	**Command:** GET EDID in1<CR><LF> **Return:** EDID in1 *10*<CR><LF> **Description:** Get in1 edid status, and the status is Fixed 3840x2160@60 PCM Audio with SDR
4	Write EDID content to an input	**Command:** SET EDID_W *in prm1 prm2*<CR><LF> **Return:** EDID_W *in prm1 prm3*<CR><LF> **Parameter:** *in* = {in1, in2...in8}; *prm1* = {block0, block1}; *prm2* = one block of 256 bytes edid ascii data with spaces (hex data need conversion into ASCII code) *prm3* = {ok, error}; error : check sum error **Description:** Write EDID content to one input.	**Command:** SET EDID_W in1 *block0 XX…XX*<CR><LF> **Return:** EDID_W in1 *block0 ok*<CR><LF> **Description:** Write EDID content to input1.
5	Read an output EDID	**Command:** GET EDID_R *out*<CR><LF> **Return:** EDID_R *out prm1 prm2*<CR><LF> **Parameter:** *out* = {out1, out2...out8}; *prm1* = {block0, block1}; *prm2* ={ one block of 256 bytes edid ascii data with no spaces(hex data need conversion into ASCII code), error, unconnect}; **Description:** Read EDID content of one output.	**Command:** GET EDID_R out1<CR><LF> **Return:** EDID_R out1 *block0 XX…XX*<CR><LF> EDID_R out1 *block1 XX…XX*<CR><LF> **Description:** EDID_R out1 *block0 XX…XX*<CR><LF> --- Read EDID ok or EDID_R *out1 error*<CR><LF> --- Check Sum Error or EDID_R out1 *unconnect*<CR><LF> --- Sink unconnect

No.	Description	Command	Example
		System Information	
1	Factory reset	**Command:** RESET<CR><LF> **Return:** RESET<CR><LF> **Description:** Factory reset	**Command:** RESET<CR><LF> **Return:** RESET<CR><LF> **Description:** Factory reset all board
2	System reboot	**Command:** REBOOT<CR><LF> **Return:** REBOOT<CR><LF> **Description:** system reboot.	**Command:** REBOOT<CR><LF> **Return:** REBOOT<CR><LF> **Description:** System reboot.
3	Get the API list	**Command:** help<CR><LF> **Return:** xxxx **Description:** Get the API list.	**Command:** help<CR><LF> **Return:** xxxx **Description:** Get the API list.
4	GET IP address	**Command:** GET IPADDR<CR><LF> **Return:** IPADDR xx.xx.xx.xx<CR><LF> **Description:** GET IP address.	**Command:** GET IPADDR<CR><LF> **Return:** IPADDR 192.168.11.243<CR><LF>
5	Get firmware version	**Command:** GET VER<CR><LF> **Return:** VER *prm*<CR><LF> **Parameter:** *prm* = {…}// according to actual firmware version **Description:** Get firmware version.	**Command:** GET VER<CR><LF> **Return:** ARM_Vx.y.z<CR><LF> **Description:** Get all module firmware version.
6	Set standby mode to on/off	**Command:** SET STANDBY prm<CR><LF> **Return:** STANDBY prm<CR><LF> **Parameter:** *prm* = {ON/OFF}// ON indicates entering low-power mode, and OFF indicates exiting low-power mode. **Description:** Set standby mode to on/off.	**Command:** SET STANDBY ON<CR><LF> **Return:** STANDBY ON<CR><LF> **Description:** Enter low-power mode

No.	Description	Command	Example
7	Get if the device is in standby mode	**Command:** GET STANDBY<CR><LF> **Return:** STANDBY prm<CR><LF> **Parameter:** *prm* = {ON/OFF}// ON indicates entering low-power mode, and OFF indicates exiting low-power mode. **Description:** Get if the device is in standby mode.	**Command:** GET STANDBY<CR><LF> **Return:** STANDBY ON<CR><LF> **Description:** Get if the device is in standby mode, the result is on.
8	Set LCD panel lock status	**Command:** SET BTN_LOCK prm<CR><LF> **Return:** BTN_LOCK prm[<CR><LF> **Parameter:** *prm* = {ON/OFF}//ON indicates that the LCD panel is locked; OFF indicates that the LCD panel lock is released. **Description:** Set LCD panel lock status.	**Command:** SET BTN_LOCK ON<CR><LF> **Return:** BTN_LOCK ON<CR><LF> **Description:** the LCD panel is locked.
9	Get LCD panel lock status	**Command:** GET BTN_LOCK prm<CR><LF> **Return:** BTN_LOCK prm<CR><LF> **Parameter:** prm = {ON/OFF}//ON indicates that the LCD panel is locked; OFF indicates that the LCD panel lock is released. **Description:** Get LCD panel lock status.	**Command:** SET BTN_LOCK ON<CR><LF> **Return:** BTN_LOCK ON<CR><LF> **Description:** the LCD panel is locked.
Preset			

No.	Description	Command	Example
1	Save or load Preset Scene	**Command:** SET PRESET_X {ACTION} {TYPE} PRESETID<CR><LF> **Return:** PRESET_X {ACTION} {TYPE} PRESETID<CR><LF> **Parameter:** PRESETID = {1, 2, 3, …}; // Preset scene number ACTION = {save|restore}; // operation type: save: Saves the current configuration to the preset scene with the specified number restore: Restores the preset scene configuration with the specified number. the {type} parameter has no effect during a restore operation. This is designed to simplify usage. TYPE = {video|audio|usb|global|videowall|multiview}; // Preset scene type video: Applies to video settings only audio: Applies to audio settings only usb: Applies to USB-related settings videowall: Applies to videowall settings only multiview: Applies to multiview settings only global: Applies to all types (video, audio, USB, etc.) **Description:** This command allows performing preset operations on a specified type of configuration (video, audio, USB, or global). According to the ACTION parameter, you can save, restore the configuration preset.	**1** **Command:** SET PRESET_X save video 1<CR><LF> **Return:** PRESET_X save video 1<CR><LF> **Description:** Save current video settings to preset scene 1. **2** **Command:** SET PRESET_X restore global 2<CR><LF> **Return:** PRESET_X restore global 2<CR><LF> **Description:** Restore the preset settings saved in preset scene 2.
2	Name Preset Scene	**Command:** SET PRESET_X name global PRESETID [NAME] <CR><LF> **Return:** PRESET_X name global PRESETID [NAME] <CR><LF> **Parameter:** PRESETID = {1, 2, 3, …}; // Preset scene number NAME = {name}; // NAME name the preset scene for name: The label can use numbers, letters, space and "_", "-"special character. Label length should be less than 32 characters. **Description:** This command allows performing preset operations on a specified type of configuration (video, audio, USB, or global). According to the ACTION parameter, you can name the configuration preset scene's name.	**3** **Command:** SET PRESET_X name global 3 MeetingRoom<CR><LF> **Return:** PRESET_X name global 3 MeetingRoom<CR><LF> **Description:** Set the name of preset scene 4 to "MeetingRoom".

No.	Description	Command	Example
3	Get Preset Status	**Command:** GET PRESET_X {prm1} {TYPE} {index_or_all} <CR><LF> **Return:** PRESET_X {prm1} {TYPE} PRESETINDEX STATUS PRESETNAME <CR><LF> **Parameter:** {prm1} = The requested operation, currently only "info" is supported. TYPE = {all}; all: Query all types (video, audio, usb, global, videowall, multiview) index_or_all = {1, 2, 3, …, all}; // Preset scene number or 'all' indicates retrieving information for all preset scenes. PRESETINDEX = Preset scene number STATUS = {0, 1, 2}; // 0 indicates an empty preset (Empty)，1 indicates a defined but inactive preset (Inactive), 2 indicates a defined and active preset (Active) PRESETNAME = Preset scene name. **Description:** This command retrieves information about preset scenes by type and index. When TYPE is 'all', it queries all preset types (video, audio, usb, global). When index_or_all is 'all', it retrieves all presets under that TYPE.	**1** **Command:** GET PRESET_X info all 1<CR><LF> **Return:** PRESET_X info video 1 2 MeetingRoom<CR><LF> **Description:** Get the information of preset scene 1. The status is 2, type is video, meaning this preset is defined and currently active, and the preset name is MeetingRoom. **2** **Command:** GET PRESET_X info all all<CR><LF> **Return:** PRESET_X info video 1 2 MeetingRoom<CR><LF> PRESET_X info audio 2 0 default_name<CR><LF> PRESET_X info usb 3 1 Conference<CR><LF> ** default_name: preset x **Description:** Query all presets from all types. Useful for full inspection or preset listing.
Output Resolution			
1	Set output resolution	**Command:** SET VIDOUT_RES *out prm*<CR><LF> **Return:** VIDOUT_RES *out prm*<CR><LF> **Description:** This function can only be set in SCALE manual mode *out = {out1~out8};* *prm =* { AUTO, BYPASS, 3840x2160@60, 3840x2160@50, 3840x2160@30, 3840x2160@25, 3840x2160@24, 1920x1200@60, 1920x1080@60, 1920x1080@50, 1680x1050@60, 1600x1200@60, 1600x900@60, 1440x900@60, 1366x768@60, 1360x768@60, 1280x1024@60, 1280x960@60, 1280x800@60, 1280x768@60, 1280x720@60, 1280x720@50, 1024x768@60, 800x600@60 ,	**Command:** SET VIDOUT_RES *out1 3840x2160@60*<CR><LF> **Return:** VIDOUT_RES *out1 3840x2160@60*<CR><LF> **Description:** Set out1 resolution is 3840x2160@60.

No.	Description	Command	Example
		Customized_1, Customized_2, Customized_3, }	
2	Get output resolution	**Command:** GET VIDOUT_RES *out*<CR><LF> **Return:** VIDOUT_RES *out prm*<CR><LF> **Description:** **This function can only be set in SCALE manual mode** *out = {out1~out8};* *prm*= { AUTO, ... 1280x800@60, 1280x768@60, 1024x768@60, 800x600@60 , Customized_1, Customized_2, Customized_3, }	**Command:** GET VIDOUT_RES *out1*<CR><LF> **Return:** VIDOUT_RES *out1 1280x800@60*<CR><LF> **Description:** HDMI out resolution is 1280x800@60.
3	Set customized resolution	**Command:** SET CUSTOM_RES *out prm hactive vactive [fps]* <CR><LF> **Return:** CUSTOM_RES out *prm hactive vactive fps* <CR><LF> **Parameter:** out = {out1~out8}; prm = {Customized_1~ Customized_3}; hactive = {0~7680}; vactive = {0~7680}; fps = 60 **Description:** set custom resolution. Note that the maximum transmission bandwidth for the hactive*vactive@fps combination cannot exceed 600Mhz.	**Command:** SET CUSTOM_RES Customized_1 *3840 2160 60* <CR><LF> **Return:** CUSTOM_RES *Customized_1 3840 2160 60* <CR><LF> **Description:** Set Customized_1 resolution to 3840x2160@60.
4	Get customized resolution	**Command:** GET CUSTOM_RES *prm* <CR><LF> **Return:** CUSTOM_RES *prm hactive vactive fps* <CR><LF> **Parameter:** out = {out1~out8}; prm = {Customized_1~Customized_3}; hactive = {0~7680}; vactive = {0~7680}; fps = 60 **Description:** Get customized resolution.	**Command:** GET CUSTOM_RES out1 Customized_1 <CR><LF> **Return:** CUSTOM_RES *out1 Customized_1 3840 2160 60* <CR><LF> **Description:** The Customized_1 resolution is 3840x2160@60.

No.	Description	Command	Example
		HTTPS & TELNETS	
1	Set HTTPS service status	**Command:** SET HTTPS prm<CR><LF> **Return:** HTTPS prm <CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Whether to enable HTTPS service.	**Command:** SET HTTPS on<CR><LF> **Return:** HTTPS on<CR><LF> **Description:** Enable HTTPS service.
2	Get HTTPS service status	**Command:** GET HTTPS<CR><LF> **Return:** HTTPS<CR><LF> **Parameter:** *prm*= {on, off}; **Description:** Get HTTP service status.	**Command:** GET HTTPS<CR><LF> **Return:** HTTPS on<CR><LF> **Description:** Get HTTP service status.
3	Set Telnet-TLS service	**Command:** SET TELNETS prm<CR><LF> **Return:** TELNETS prm <CR><LF> **Parameter:** *prm*= {on, off};// on means to enable **Description:** Telnet-TLS; off means to enable Telnet. TELNETS is short for Telnet-TLS.	**Command:** SET TELNETS on<CR><LF> **Return:** TELNETS on<CR><LF> **Description:** Enable Telnet-TLS.
4	Get TelnetS service status	**Command:** GET TELNETS<CR><LF> **Return:** TELNETS prm <CR><LF> **Parameter:** *prm*= {on, off};// on means to enable **Description:** Telnet-TLS; off means to enable Telnet. TELNETS is short for Telnet-TLS.	**Command:** GET TELNETS<CR><LF> **Return:** TELNETS on<CR><LF> **Description:** To get Telnet service status, and the result is Telnet-TLS is enabled.
5	Sets TelnetS New password	**Command:** SET TEL_PWD Oldpwd Newpwd NewpwdVerify<CR><LF> **Return:** TEL_PWD prm<CR><LF> **Parameter:** Old/Newpwd : Must be 4 to 16 characters in lentgh (alphanumeric only) prm : {0, 1}; // 0: failed, 1: success **Description:** Set the username and password for logging in to the telnet.	**Command:** SET TEL_PWD admin admin123 admin123<CR><LF> **Return:** TEL_PWD 1<CR><LF> **Description:** Set Telnet's password [admin] to [admin123]

No.	Description	Command	Example
		Video / Audio Detail	
1	Get the connection status of the video input	**Command:** GET VIDIN_CONNECT *in*<CR><LF> **Return:** VIDIN_CONNECT *in prm*<CR><LF> *in* = {in1, in2...in8}; *prm* = {DISCONNECTE, CONNECTE} **Description:** Get the connection status of the video input.	**Command:** GET VIDIN_CONNECT in1<CR><LF> **Return:** VIDIN_CONNECT in1 DISCONNECTE<CR><LF>
2	Get the signal status of the video input	**Command:** GET VIDIN_SIG *in*<CR><LF> **Return:** VIDIN_SIG *in prm*<CR><LF> *in* = {in1, in2...in8}; *prm* = {No, Valid} **Description:** Get the signal status of the video input.	**Command:** GET VIDIN_SIG in1<CR><LF> **Return:** VIDIN_SIG in1 Valid<CR><LF>
3	Get input video format information	**Command:** GET VIDIN_FORMAT *in*<CR><LF> **Return:** VIDIN_FORMAT i*n prm*<CR><LF> in = {in1, in2...in8}; prm = { } **Description:** Get the resolution of the video input prm = {<horizontal>x<vertical>,<rate>; <HDR info>;<ColorSpace>,<DeepColor>} • horizontal = An integer value representing the horizontal. • vertical = An integer value representing the vertical. May have an additional qualifier such as 'i' or 'p'. • rate = An integer value representing the refresh rate. • HDR info = none hdr/ static hdr/ dynamic hdr • Color space = RGB / Ycbcr 444 /Ycbcr 422/Ycbcr 420 • DeepColor = 8 bit/10 bit /12 bit/ 16 bit	**Command:** GET VIDIN_FORMAT *in1*<CR><LF> **Return:** VIDIN_FORMAT *in1 3840*x2160,60;none hdr;rgb;8bit<CR><LF>
4	Get input audio format information	**Command:** GET AUDIN_FORMAT *in*<CR><LF> **Return:** AUDIN_FORMAT i*n prm*<CR><LF> in = {in1,in2...in8}; prm = { } **Description:** Get the resolution of the video input prm = {<PCM / non-PCM>; <Sampling rate>}	**Command:** GET AUDIN_FORMAT *in1*<CR><LF> **Return:** AUDIN_FORMAT *in1 PCM*;48kHZ<CR><LF>
5	Get hdcp version of input video	**Command:** GET VIDIN_HDCP *in*<CR><LF> **Return:** VIDIN_HDCP *in prm*<CR><LF> in = {in1,in2...in8}; prm = {no hdcp, hdcp1.4, hdcp2.2} **Description:** Get hdcp version of input video	**Command:** GET VIDIN_HDCP *in1*<CR><LF> **Return:** VIDIN_HDCP *in1 no hdcp*<CR><LF>

No.	Description	Command	Example
6	Get the connection status of the video output	**Command:** GET VIDOUT_CONNECT *out*<CR><LF> **Return:** VIDOUT_CONNECT *out prm*<CR><LF> out = {out1,…,out8}; *prm* = {DISCONNECTE, CONNECTE} **Description:** Get the connection status of the video output	**Command:** GET VIDOUT_CONNECT out1<CR><LF> **Return:** VIDOUT_CONNECT out1 DISCONNECTE<CR><LF>
7	Get the signal status of the video output	**Command:** GET VIDOUT_SIG *out*<CR><LF> **Return:** VIDOUT_SIG *out prm*<CR><LF> *out* = {out1, out2…out8}; *prm* = {No, Valid} **Description:** Get the signal status of the video output.	**Command:** GET VIDOUT_SIG out1<CR><LF> **Return:** VIDOUT_SIG *out1 Valid*<CR><LF>
8	Get the video format information of output	**Command:** GET VIDOUT_FORMAT out<CR><LF> **Return:** VIDOUT_FORMAT out prm<CR><LF> out = {out1, out2…out8}; prm = { } **Description:** Get the resolution of the video input prm = {<horizontal>x<vertical>,<rate>; <HDR info>;<ColorSpace>,<DeepColor>} • horizontal = An integer value representing the horizontal. • vertical = An integer value representing the vertical. May have an additional qualifier such as 'i' or 'p'. • rate = An integer value representing the refresh rate. • HDR info = none hdr/ static hdr/ dynamic hdr • Color space = RGB / Ycbcr 444 /Ycbcr 422/Ycbcr 420 • DeepColor = 8 bit/10 bit /12 bit/ 16 bit	**Command:** GET VIDOUT_FORMAT out1<CR><LF> **Return:** VIDOUT_FORMAT out1 3840x2160,60;none hdr;rgb;8bit<CR><LF>
9	Get the audio format information of output	**Command:** GET AUDOUT_FORMAT in<CR><LF> **Return:** AUDOUT_FORMAT in prm<CR><LF> out = {out1,out2…out8}; prm = { } **Description:** Get the resolution of the video input prm = {<PCM / non-PCM>; <Sampling rate>}	**Command:** GET AUDOUT_FORMAT out1<CR><LF> **Return:** AUDOUT_FORMAT out1 PCM;48kHZ<CR><LF>
10	Get hdcp version of output video	**Command:** GET VIDOUT_HDCP out<CR><LF> **Return:** VIDOUT_HDCP out prm<CR><LF> out = {out1, out2…out8}; prm = {no hdcp, hdcp1.4, hdcp2.2} **Description:** Get hdcp version of output video.	**Command:** GET VIDOUT_HDCP out1<CR><LF> **Return:** VIDOUT_HDCP out1 no hdcp<CR><LF>

No.	Description	Command	Example
		Stretch	
1	Configure the HDMI output image to best fit or full	**Command:** SET VIDOUT_STRETCH out prm<CR><LF> **Return:** VIDOUT_STRETCH out prm<CR><LF> **Parameter:** *out = {out1,out2…out8};* *prm = {0, 1}; //0: full, 1: bestfit* **Description:** Set Stretch Mode.	**Command:** SET VIDOUT_STRETCH out1 0<CR><LF> **Return:** VIDOUT_STRETCH out1 0<CR><LF> **Description:** Set out1's Stretch Mode to full.
2	Get stretch mode of the output image	**Command:** GET VIDOUT_STRETCH out prm<CR><LF> **Return:** VIDOUT_STRETCH out prm<CR><LF> **Parameter:** out = {out1,out2…out8}; prm = {0, 1}; //0: full, 1: bestfit **Description:** Get Stretch Mode	**Command:** GET VIDOUT_STRETCH out1<CR><LF> **Return:** VIDOUT_STRETCH out1 0<CR><LF> **Description:** Get out1's Stretch Mode, the result is full.