AT-OME-CS31-SA

AT-OME-CS31-SA-C

AT-OME-CS31-SA-HDBT

Omega™ 3x1 Switcher Mixer Amp

API Command Reference

October 2026



Table of Contents

Version Information....3
Sales & Customer Support....3
Introduction....4
Connection Information....4
TCP Proxy....4
Command Structure....4
Command Quick Reference....5
Audio....5
Display (CEC/RS-232/IP)....5
EDID....6
HDCP....6
Network....6
Occupancy Sensor....6
PA Sense....7
System....7
USB....7
Video....8
Commands....9
Audio....9
Display (CEC/RS-232/IP)....25
EDID....33
HDCP....37
Network....39
Occupancy Sensor....44
PA Sense....48
System....51
USB....55
Video....59

Version Information

This document lists the commands for controlling the AT-OME-CS31-SA, AT-OME-CS31-SA-C, AT-OME-CS31-SA-HDBT over Telnet, SSH, WebSocket (JSON-RPC), and RS-232. Its version history follows the original command reference.

Version	Date	Notes
1	Jun 2024	Combined API (SA, SA-HDBT, and SA-C)
HR 1.0	October 2026	Converted to the Hall Research document format. Firmware 1.0.2: LumaStream stream decoding added (VideoStreamNameList.Get and SelectVideoStream.Set; in4 = Stream input on VideoSwitch.Set, VideoSwitchDefault.Set, and VideoInputStatus.Get).
HR 1.1	October 2026	Command descriptions and parameter tables corrected from the original command reference.

Sales & Customer Support

Item	Details
Main Office (US)	Hall Research, 1234 Lakeshore Dr Suite #150, Coppell, TX 75019
Support Hours	Monday – Friday, 8:00 a.m. – 6:00 p.m. (CST)
Website	https://www.hallresearch.com
Product Page (AT-OME-CS31-SA)	https://www.hallresearch.com/product/AT-OME-CS31-SA
Product Page (AT-OME-CS31-SA-C)	https://www.hallresearch.com/product/AT-OME-CS31-SA-C
Product Page (AT-OME-CS31-SA-HDBT)	https://www.hallresearch.com/product/AT-OME-CS31-SA-HDBT

IMPORTANT: Visit hallresearch.com/product/AT-OME-CS31-SA or hallresearch.com/product/AT-OME-CS31-SA-C or hallresearch.com/product/AT-OME-CS31-SA-HDBT for the latest firmware updates and documentation.

Introduction

This document applies to the AT-OME-CS31-SA, AT-OME-CS31-SA-C, and AT-OME-CS31-SA-HDBT. All three share the same command set (firmware 1.0.2). The Stream input (in4) is available on all three models from firmware 1.0.2. Input numbers refer to each model's own inputs.

Model	Inputs (in1, in2, in3)
AT-OME-CS31-SA	Three HDMI inputs
AT-OME-CS31-SA-C	One USB-C input and two HDMI inputs
AT-OME-CS31-SA-HDBT	One HDBaseT input and two HDMI inputs

Connection Information

Item	Setting
Telnet port	23
SSH port	22
WebSocket (WS)	Port 80 — ws://<IP>/ws
Secure WebSocket (WSS)	Port 443 — wss://<IP>/ws
RS-232 settings (default)	115200 baud, 8 data bits, no parity, 1 stop bit

TCP Proxy

TCP Proxy passes data to and from the RS-232 ports over a TCP socket, using the following port assignments:

TCP Port	Description
9001	RS-232 port 1
9002	RS-232 port 2

NOTE: Port 9000 is not supported. TCP Proxy is not supported on the AT-OME-CS31-SA-HDBT HDBaseT IN port.

Command Structure

Each command uses the JSON-RPC 2.0 structure: {"jsonrpc": "2.0", "id": "X", "method": "...", "params": ...}. The id is optional but recommended; it is returned with the response, so responses can be matched to the commands that produced them. Commands are case-sensitive.

NOTE: Each command and each response is a single line when sent over Telnet or SSH. Where a command wraps on the page, the wrap is not part of the command. Responses are shown indented for readability.

Command Quick Reference

Select a command name to go to its full description.

Audio

Command	Description
AmplifierMode.Get	Returns the audio mode of amplifier output
AmplifierMode.Set	Sets the amplifier audio output mode to stereo or mono
AudioDuck.Get	Returns the audio ducking status
AudioDuckAttack.Set	Sets the ducking attack time, in milliseconds
AudioDuckDepth.Set	Sets the ducking depth
AudioDuckEn.Set	Enables or disables audio ducking
AudioDuckRelease.Set	Sets the ducking release time, in milliseconds
AudioDuckSrc.Set	Sets which source has a higher priority, when triggering audio ducking
AudioDuckThreshold.Set	Sets the ducking threshold, in decibels
AudioInput.Get	Returns the audio input status
AudioInputFormat.Get	Returns input audio format information
AudioInputMute.Set	Enables or disables audio input muting
AudioInputVol.Set	Sets the volume for the audio input, in decibels
AudioOutFormat.Get	Returns the audio format of the output
AudioOutput.Get	Returns the audio output mute status
AudioOutputFollow.Set	Enables or disables the linking of balanced audio out and amplifier audio out
AudioOutputMute.Set	Mutes the audio output
AudioOutputVol.Set	Sets audio output volume, in decibels

Display (CEC/RS-232/IP)

Command	Description
DisplayCtrl.Get	Returns the display control mode
DisplayCtrlCmd.Set	Sets the command string to send to the display
DisplayCtrlDelay.Set	Sets the time interval, in minutes, before the command to power-off the display is sent, when an A/V signal is no longer present
DisplayCtrlIP.Set	Specifies the IP settings for the display
DisplayCtrlMode.Set	Sets the protocol to be used to control the display

Command	Description
DisplayCtrlOperation.Set	Controls the display
DisplayCtrlRs232.Set	Sets the RS-232 parameters for communicating with the display over RS-232
DisplayPowerOnAuto.Set	Enables or disables turning on/off the display automatically, according to the output signal on HDMI output port

EDID

Command	Description
EDIDInput.Get	Returns the EDID input status
EDIDInput.Set	Sets the EDID on the specified input
EDIDInputFile.Set	Sets an EDID file to input EDID
EdidSinkFile.Get	Fetches the EDID from the sink that is connected to the output

HDCP

Command	Description
HdcpCompliant.Get	Returns the current HDCP state of all inputs
HdcpCompliant.Set	Sets the HDCP state for the desired input(s)

Network

Command	Description
Network.Get	Returns the current network settings of the unit
Network.Set	Sets the network settings of the unit
NetworkCtrlProtocols.Get	Returns the network control protocol
NetworkCtrlProtocolsEnable.Set	Enables or disables the desired network protocol
NetworkHostname.Get	Returns the hostname of the unit
NetworkHostname.Set	Sets the hostname of the unit

Occupancy Sensor

Command	Description
OccSensor.Get	Returns the status of the occupancy sensor
OccSensorAdd	Adds the JAV-OCS-900N as a sensor

Command	Description
OccSensorDelete	Removes the specified JAV-OCS-900N sensor
OccSensorIdentify	Enabled the blinking of the LED indicators on the JAV-OCS-900N to physically identify the unit
OccSensorSettings	Enables or disables the standard I/O and Atlona IP sensors

PA Sense

Command	Description
PASense.Get	Returns the PA sense system release delay
PASenseRelease.Set	Sets PA sense system release delay, in seconds
PASenseTriggerMode.Set	Sets the trigger port for PA sense system

System

Command	Description
Platform.FactoryReset	Resets the unit to factory default settings
Platform.Reboot	Reboots the unit
System.Get	Returns information about the unit
SystemBlinkLed.Set	Enables or disables blinking of the LED indicators on the front panel
SystemFrontPanel.Set	Enables or disables the front panel buttons
SystemStandby.Set	Places the unit in standby mode

USB

Command	Description
UsbFollowVideoCfg.Set	Assigns the USB host port to video ports
UsbSwitch.Get	Returns the current USB switch setting
UsbSwitchHost.Set	Sets which USB Host is switched to the USB device in manual switching mode
UsbSwitchMode.Set	Sets the USB switching mode
UsbVbus.Set	Toggles the Vbus

Video

Command	Description
SelectVideoStream.Set	Selects the LumaStream stream to decode
VideoAutoSw.Set	Enables or disables auto switching
VideoHDBTStatus.Get	Returns the HDBaseT status
VideoHDMIOut5V.Get	Returns the status of HDMI +5 V output when no output signal is present
VideoHDMIOut5V.Set	Enables or disables the HDMI +5 V when no output signal is present
VideoInputStatus.Get	Returns the status of the inputs
VideoOutputStatus.Get	Returns the status of the output
VideoStreamNameList.Get	Returns the LumaStream video streams available to the switcher
VideoSwitch.Get	Returns the current switching settings
VideoSwitch.Set	Switches the specified input to the output
VideoSwitchDefault.Set	Assigns the default input to be switched to, when no video is present on the last connected input, in auto switching mode

Commands

Audio

AmplifierMode.Get

Returns the audio mode of amplifier output.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AmplifierMode.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AmplifierMode.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "amplifiermode": "stereo"
    },
    "jsonrpc": "2.0"
}
```

AmplifierMode.Set

Sets the amplifier audio output mode to stereo or mono.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AmplifierMode.Set","params":"Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	stereo, mono

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AmplifierMode.Set", "params": "stereo" }
```

Response

```
{
    "id": "User1",
    "jsonrpc": "2.0",
    "result": {
        "amplifiermode": "stereo"
    }
}
```

AudioDuck.Get

Returns the audio ducking status.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuck.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuck.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/enable": "true",
        "AudioDuck/source": "audio1",
        "AudioDuck/threshold": -30,
        "AudioDuck/depth": -30,
        "AudioDuck/attack": 100,
        "AudioDuck/release": 100
    },
    "jsonrpc": "2.0"
}
```

AudioDuckAttack.Set

Sets the ducking attack time, in milliseconds.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckAttack.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	1...1000

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckAttack.Set", "params": 100 }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/attack": 100
    },
    "jsonrpc": "2.0"
}
```

AudioDuckDepth.Set

Sets the ducking depth. Depth is specified in decibels.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckDepth.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	-60...0

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckDepth.Set", "params": -30 }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/depth": -30
    },
    "jsonrpc": "2.0"
}
```

AudioDuckEn.Set

Enables or disables audio ducking.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckEn.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckEn.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/enable": "true"
    },
    "jsonrpc": "2.0"
}
```

AudioDuckRelease.Set

Sets the ducking release time, in milliseconds.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckRelease.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	1...3000

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckRelease.Set", "params": 100 }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/release": 100
    },
    "jsonrpc": "2.0"
}
```

AudioDuckSrc.Set

Sets which source has a higher priority, when triggering audio ducking. When audio is present on both the balanced audio input and the H.264/265 input, the input designated as the highest priority will remain at its set level, and the other input will be ducked along with all other audio inputs.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckSrc.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	audio1 // balanced input audio2 // H.264/H.265 input

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckSrc.Set", "params": "audio1" }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/source": "audio1"
    },
    "jsonrpc": "2.0"
}
```

AudioDuckThreshold.Set

Sets the ducking threshold, in decibels.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioDuckThreshold.Set", "params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	-60...0

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioDuckThreshold.Set", "params": -30 }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioDuck/threshold": -30
    },
    "jsonrpc": "2.0"
}
```

AudioInput.Get

Returns the audio input status.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioInput.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioInput.Get" }
```

Response


```
{
    "id": "User1",
    "jsonrpc": "2.0",
    "result": {
        "balanced/signal": "NO",
        "localmix/HDMI/muted": false,
        "localmix/HDMI/volume": 0,
        "localmix/USB/muted": false,
        "localmix/USB/volume": 0,
        "localmix/balanced/muted": false,
        "localmix/balanced/volume": 0,
        "localmix/unbalanced1/muted": false,
        "localmix/unbalanced1/volume": 0,
        "localmix/unbalanced2/muted": false,
        "localmix/unbalanced2/volume": 0,
        "remotemix/HDMI/muted": false,
        "remotemix/HDMI/volume": 0,
        "remotemix/MIC/muted": false,
        "remotemix/MIC/volume": 0,
        "remotemix/balanced/muted": false,
        "remotemix/balanced/volume": 0,
        "remotemix/unbalanced1/muted": false,
        "remotemix/unbalanced1/volume": 0,
        "remotemix/unbalanced2/muted": false,
        "remotemix/unbalanced2/volume": 0,
        "unbalanced1/signal": "NO",
        "unbalanced2/signal": "NO"
    }
}
```

AudioInputFormat.Get

Returns input audio format information.

Syntax

```
{ "jsonrpc": "2.0","id": "X","method": "AudioInputFormat.Get","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioInputFormat.Get", "params": "in2" }
```

Response

```
{
    "id": "User1",
    "result": {
        "source": "in2",
        "audioformat": "PCM;48kHZ"
    },
    "jsonrpc": "2.0"
}
```

AudioInputMute.Set

Enables or disables audio input muting. Note that USB can only be selected by the local mixer, and MIC can only be selected by the remote mixer.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioInputMute.Set","params": {"mixer": "Y","source": "Z","mute": W}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	local, remotemix
Z	params	balanced, unbalanced1, unbalanced2, HDMI, USB, MIC
W	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioInputMute.Set", "params": { "mixer": "localmix", "source": "USB", "mute": true } }
```

Response

```
{
    "id": "User1",
    "result": {
        "mixer": "localmix",
        "source": "USB",
        "mute": true
    },
    "jsonrpc": "2.0"
}
```

AudioInputVol.Set

Sets the volume for the audio input, in decibels. Note that USB can only be selected by the local mixer, and MIC can only be selected by the remote mixer. up and down arguments can be used to increment or decrement the volume input level by 1, respectively.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioInputVol.Set","params":{"mixer": "Y","source": "Z","volume": W}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	local, remotemix
Z	params	balanced, unbalanced1, unbalanced2, HDMI, USB
W	params	-60...0, up, down

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioInputVol.Set", "params": { "mixer": "localmix", "source": "USB", "volume": -50 } }
```

Response

```
{
    "id": "User1",
    "result": {
        "mixer": "localmix",
        "source": "USB",
        "volume": -50
    },
    "jsonrpc": "2.0"
}
```

AudioOutFormat.Get

Returns the audio format of the output.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioOutFormat.Get","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	out1

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutFormat.Get", "params": "out1" }
```

Response

```
{
    "id": "User1",
    "result": {
        "sink": "out1",
        "audioformat": "PCM;48kHZ"
    },
    "jsonrpc": "2.0"
}
```

AudioOutput.Get

Returns the audio output mute status.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioOutput.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutput.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "mixerout/follow": false,
        "balanced/muted": false,
        "balanced/volume": -40,
        "amp/muted": false,
        "amp/volume": -40,
        "uacaudio/muted": false,
        "uacaudio/volume": -40,
        "hdmiout/muted": false
    },
    "jsonrpc": "2.0"
}
```

AudioOutputFollow.Set

Enables or disables the linking of balanced audio out and amplifier audio out. When enabled (true), the volume and the mute/unmute status of the two audio outputs can be set together. When disabled (false), the volume and the mute/unmute status of the two audio outputs can be controlled independently.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioOutputFollow.Set","params":Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutputFollow.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "AudioOutputFollow": true
    },
    "jsonrpc": "2.0"
}
```

AudioOutputMute.Set

Mutes the audio output. If the command AudioOutputFollow.Set is set to true, then the request object "audioout" can be set to balanced or amp.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioOutputMute.Set","params":{"audioout": "Y","mute": Z}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	balanced, amp, uacaudio, hdmiout
Z	params	true, false, toggle

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutputMute.Set", "params": { "audioout": "amp", "mute": true } }
```

Response

```
{
    "id": "User1",
    "result": {
        "audioout": "amp",
        "mute": true
    },
    "jsonrpc": "2.0"
}
```

AudioOutputVol.Set

Sets audio output volume, in decibels. If the command AudioOutputFollow.Set is set to true, then the request object "audioout" can be set to balanced or amp. up and down arguments can be used to increment or decrement the volume output level by 1, respectively.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "AudioOutputVol.Set","params":{"audioout": "Y","volume": Z}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	balanced, amp, uacaudio
Z	params	-80...0, up, down

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutputVol.Set", "params": { "audioout": "amp", "volume": -60 } }
```

Response

```
{
    "jsonrpc": "2.0",
    "id": "User1",
    "method": "AudioOutputVol.Set",
    "params": {
        "audioout": "amp",
        "volume": -60
    }
}
```

Display (CEC/RS-232/IP)

DisplayCtrl.Get

Returns the display control mode.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrl.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayCtrl.Get" }
```

Response


```
{
    "id": "User1",
    "result": {
        "DisplayControl/controlmethod": "RS232",
        "DisplayControl/AutocontrolEnabe": true,
        "DisplayControl/AutocontrolDelay": 2,
        "DisplayControl/ip/address": "192.168.XX.XX",
        "DisplayControl/ip/port": "1111",
        "DisplayControl/ip/mode": "Login",
        "DisplayControl/ip/username": "admin",
        "DisplayControl/ip/password": "admin",
        "DisplayControl/RS232": [
            {
                "prot ": "uart1 ",
                "Bitrate ": "9600 ",
                "Databits ": "8 ",
                "Parity ": "N ",
                "Stopbits ": "1 "
            },
            {
                "prot ": "uart2 ",
                "Bitrate ": "115200 ",
                "Databits ": "8 ",
                "Parity ": "N ",
                "Stopbits ": "1 "
            }
        ],
        "DisplayControl/command/RepeatStatus": true,
        "DisplayControl/command/RepeatTimes": "2",
        "DisplayControl/command/off": "\\x08\\ x22\\ x00\\ x00\\ x00\\ x01\\ xD5 ",
        "DisplayControl/command/on": "\\x08\\ x22\\ x00\\ x00\\ x00\\ x02\\ xD4 ",
        "DisplayControl/command/volumeup": "\\ x08\\ x22\\ x00\\ x00\\ x00\\ x01\\ x03 ",
        "DisplayControl/command/volumedown": "\\x08\\ x22\\ x00\\ x00\\ x00\\ x02\\ x03 ",
        "DisplayControl/command/mute": "\\x08\\ x22\\ x00\\ x00\\ x00\\ x01\\ x04 ",
        "DisplayControl/command/unmute": "\\x08\\ x22\\ x00\\ x00\\ x00\\ x02\\ x05 "
    },
    "jsonrpc": "2.0"
}
```

DisplayCtrlCmd.Set

Sets the command string to send to the display.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlCmd.Set","params": {"cmd": "Y","data": "W"}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	poweron, poweroff, volumeup,volumedown, mute, unmute
Z	params	[data]

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayCtrlCmd.Set", "params": { "cmd": "poweron", "data": "02 50 4f 4e 03" } }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

DisplayCtrlDelay.Set

Sets the time interval, in minutes, before the command to power-off the display is sent, when an A/V signal is no longer present.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlDelay.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	1...30

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayCtrlDelay.Set", "params": 2 }
```

Response

```
{
    "id": "User1",
    "result": {
        "displayctrldelay": 2
    },
    "jsonrpc": "2.0"
}
```

DisplayCtrlIP.Set

Specifies the IP settings for the display.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlIP.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	ipaddress	IP address of the display
Z	port	Port
M	mode	Login, Non-Login
N	username/password	String

Example

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlIP.Set","params": {"mode":"M","ipaddress":"Y","port":Z,
"username":"N","password":"N"}}
```

Response

```
{
    "jsonrpc": "2.0",
    "id": "User1",
    "method": "DisplayCtrlIP.Set",
    "params": {
        "mode": "Login",
        "ipaddress": "10.1.0.200",
        "port": 23,
        "username": "admin",
        "password": "TEST1"
    }
}
```

DisplayCtrlMode.Set

Sets the protocol to be used to control the display.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlMode.Set","params":Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	IP, RS-232, CecAPI

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayCtrlMode.Set", "params": "RS232" }
```

Response

```
{
    "id": "User1",
    "result": {
        "displaycontrolmethod": "RS232"
    },
    "jsonrpc": "2.0"
}
```

DisplayCtrlOperation.Set

Controls the display.

Syntax

```
{ "jsonrpc": "2.0", "id" "X": "DisplayCtrlOperationSetResults", "method": "DisplayCtrlOperation.Set",
"params": { "command": "Y", "controlmode": "Z" } }
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	poweron, poweroff,volumeup,volumedown, mute, unmute
Z	params	IP, RS232, CecAPI

Example

```
{ "jsonrpc": "2.0", "id": "DisplayCtrlOperationSetResults", "method": "DisplayCtrlOperation.Set", "params":
{ "command": "poweron", "controlmode": "CecAPI" } }
```

Response

```
{
    "id": "DisplayCtrlOperationSetResults",
    "jsonrpc": "2.0",
    "result": {
        "DisplayCtrlOperation": true
    }
}
```

DisplayCtrlRs232.Set

Sets the RS-232 parameters for communicating with the display over RS-232.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayCtrlRs232.Set","params": {"port": Y,"baudrate": Z,"parity": L,
"dataBit": M,"stopBit": N}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	uart1, uart2
Z	params	9600, 19200, 38400, 57600, 115200
L	params	N, O, E
M	params	7, 8
N	params	1, 2

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayCtrlRs232.Set", "params": { "port": "uart1", "baudrate":
"115200", "parity": "N", "dataBit": "8", "stopBit": "1" } }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

DisplayPowerOnAuto.Set

Enables or disables turning on/off the display automatically, according to the output signal on HDMI output port.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "DisplayPowerOnAuto.Set","params":Y }
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "DisplayPowerOnAuto.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "displaypowonauto": true
    },
    "jsonrpc": "2.0"
}
```

EDID

EDIDInput.Get

Returns the EDID input status.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "EDIDInput.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "EDIDInput.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "mem1": true,
        "mem1EDID ": "xxxx",
        "edidsettings": [
            {
                "source": "in1",
                "edidmode": 2
            },
            {
                "source": "in2",
                "edidmode": 2
            },
            {
                "source": "in3",
                "edidmode": 2
            }
        ]
    },
    "jsonrpc": "2.0"
}
```

EDIDInput.Set

Sets the EDID on the specified input.

The values for the third parameter refer to the following:

1 = Copy from the hdmi output connected display (Default).

2 = 3840x2160 @ 60 Hz 2.0 CH PCM Audio with HDR.

3 = 3840x2160 @ 30 Hz 2.0 CH PCM Audio with SDR.

4 = 1920x1080P @ 60 Hz 2.0 CH PCM Audio with SDR.

5 = 1920x1080P @ 30 Hz 2.0 CH PCM Audio with SDR.

6 = 1280x720P @ 60 Hz 2.0 CH PCM Audio with SDR.

7 = 1280x800 2.0 CH PCM Audio with SDR.

8 = 1920x1200 2.0 CH PCM Audio with SDR.

9 = 5120x2160 @ 30 Hz 2.0 CH PCM Audio with SDR.

10 = 3440x1440 @ 50 Hz 2.0 CH PCM Audio with SDR.

11 = 2560x1080 @ 60 Hz 2.0 CH PCM Audio with SDR.

99 = Custom.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "EdidInput.Set","params": {"source": "Y","edidmode": Z}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3, all
Z	params	1...11, 99

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "EdidInput.Set", "params": { "source": "in1", "edidmode": 2 } }
```

Response

```
{
    "id": "User1",
    "result": {
        "source": "in1",
        "edidmode": 2
    },
    "jsonrpc": "2.0"
}
```

EDIDInputFile.Set

Sets an EDID file to input EDID. EDID data must be a hexadecimal string with no spaces.

Syntax

{"jsonrpc": "2.0","id": "X","method": "EdidInputFile.Set","params": {"edid": "Y"}}

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	[512 bytes hexadecimal EDID data]

Example

{"jsonrpc": "2.0","id": "User1","method": "EdidInputFile.Set","params": {"source": "all, "edid": "
2000000000FFFFFFFFFFFF004C2DCB0B000000000D180103804728782AD691A7554EA0256720002000
0C5054BFEF80714F810081C081809500A9C0B3000101565E00A0A0A0295030204F200040003500C48E2
100001A023A801871382D40582C4500C48E2100001E000000FD00188520006000 4B1B5A19000A202020
202020000000FC00533332443835300A2020202020011885200080000 2031CF14890041F1303122022230
907078301000066030C00100080023A8018522000A00071382D40582C4500C48E2100001E023A80D0723
82D40102C4580C48E2100001E9B2000C000011D007251D01E206E285500C48E2100001E011D00BC52D
01E20B8285540C48EB42000E0002100001E000000000000000000000000000000000000000000000000000
00000A21F00000001FF"}}

Response

{
    "id": "User1",
    "result": {
        "edid":
"2000000000FFFFFFFFFFFF004C2DCB0B000000000D180103804728782AD691A7554EA0256720002000
0C5054BFEF80714F810081C081809500A9C0B3000101565E00A0A0A0295030204F200040003500C48E2
100001A023A801871382D40582C4500C48E2100001E000000FD00188520006000 4B1B5A19000A202020
202020000000FC00533332443835300A2020202020011885200080000 2031CF14890041F1303122022230
907078301000066030C00100080023A8018522000A00071382D40582C4500C48E2100001E023A80D072
382D40102C4580C48E2100001E9B2000C000011D007251D01E206E285500C48E2100001E011D00BC52
D01E20B8285540C48EB42000E0002100001E000000000000000000000000000000000000000000000000000
000000A21F00000001FF"
    },
    "jsonrpc": "2.0"
}

EdidSinkFile.Get

Fetches the EDID from the sink that is connected to the output.

Syntax

```
{ "jsonrpc": "2.0","id": "X","method": "EdidSinkFile.Get","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	out1

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "EdidSinkFile.Get", "params": "out1" }
```

Response


```
{
    "id": "User1",
    "result": {
        "edid":
"00ffffffffffff004c2d7c0b0000000033170103805932780aee91a3544c99260f5054bdef80714f8
1c0810081809500a9c0b3000101023a801871382d40582c450076f23100001e662156aa51001e30468f3300
76f23100001e000000fd00184b0f5117000a2020202020000000fc0053414d53554e470a202020202001e0
020325f14d901f0413051403122021220716230907078301000e2000f67030c001000b82d011d80d0721c1
620102c258076f23100009e011d8018711c1620582c250076f23100009e011d00bc52d01e20b828554076f2
3100001e011d007251d01e206e28550076f23100001e000000000000000000000000000000000000070"
    },
    "jsonrpc": "2.0"
}
```

HDCP

HdcpCompliant.Get

Returns the current HDCP state of all inputs.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "HdcpCompliant.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "HdcpCompliant.Get" }
```

Response

```
{
    "id": "User1",
    "jsonrpc": "2.0",
    "result": {
        "hdcpCompliants": [
            {
                "hdcpCompliant": true,
                "source": "in1"
            },
            {
                "hdcpCompliant": true,
                "source": "in2"
            },
            {
                "hdcpCompliant": true,
                "source": "in3"
            }
        ]
    }
}
```

HdcpCompliant.Set

Sets the HDCP state for the desired input(s).

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "HdcpCompliant.Set","params": {"source": "Y","hdcpCompliant": S}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3, all
Z	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "HdcpCompliant.Set", "params": { "source": "in1",
"hdcpCompliant": true } }
```

Response

```
{
    "id": "User1",
    "result": {
        "source": "in1",
        "hdcpCompliant": true
    },
    "jsonrpc": "2.0"
}
```

Network

Network.Get

Returns the current network settings of the unit.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "Network.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Network.Get" }
```

Response

```
{
    "id": "User1",
    "jsonrpc": "2.0",
    "result": {
        "gateway": "10.1.1.254",
        "ip_mode": "dhcp",
        "ip_port": 23,
        "ipaddr": "10.1.0.26",
        "mac": "B8:98:B0:0E:F4:63",
        "netmask": "255.255.254.0"
    }
}
```

Network.Set

Sets the network settings of the unit.

Syntax

```
{"jsonrpc": "2.0","id": "U","method": "Network.Set","params":{"ip_mode": "V","ipaddr": "W","netmask": "X", "gateway": "Y","ip_port": Z}}
```

Identifier	Request Object	Parameters
U	id	Enter a user id. This is optional.
V	params	static, dhcp
W	params	IP address (xxx.xxx.xxx.xxx)
X	params	Netmask (xxx.xxx.xxx.xxx)
Y	params	Gateway (xxx.xxx.xxx.xxx)
Z	params	Port number

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Network.Set", "params": { "ip_mode": "static", "ipaddr": "192.168.2.100", "netmask": "255.255.255.0", "gateway": "192.168.2.1", "ip_port": 23 } }
```

Response

```
{
    "id": "User1",
    "result": {
        "ip_mode": "static",
        "ipaddr": "192.168.2.100",
        "netmask": "255.255.255.0",
        "gateway": "192.168.2.1",
        "ip_port": 23
    },
    "jsonrpc": "2.0"
}
```

NetworkCtrlProtocols.Get

Returns the network control protocol.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "NetworkCtlProtocols.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "NetworkCtlProtocols.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "telnet": {
            "enabled": true,
            "timeout": "off"
        },
        "ssh": {
            "enabled": true,
            "timeout": "off"
        }
    },
    "jsonrpc": "2.0"
}
```

NetworkCtrlProtocolsEnable.Set

Enables or disables the desired network protocol.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "NetworkCtlProtocolsEnable.Set","params":{"protocol":"Y","enabled": Z}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	telnet, ssh
Z	params	true, false

Example

```
{"jsonrpc": "2.0","id": "User1","method": "NetworkCtlProtocolsEnable.Set","params":{"protocol":"telnet" "enabled": false}}
```

Response

```
{
    "id": "User1",
    "result": {
        "protocol": "telnet",
        "enabled": false
    },
    "jsonrpc": "2.0"
}
```

NetworkHostname.Get

Returns the hostname of the unit.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "NetworkHostname.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "NetworkHostname.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "hostname": "ATL"
    },
    "jsonrpc": "2.0"
}
```

NetworkHostname.Set

Sets the hostname of the unit. If no hostname is provided, then the default hostname will be assigned.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "NetworkHostname.Set","hostname":"Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	Hostname

Example

```
{"jsonrpc": "2.0","id": "User1","method": "NetworkHostname.Set","hostname": "ATL"}}
```

Response

```
{
    "id": "User1",
    "result": {
        "hostname": "ATL"
    },
    "jsonrpc": "2.0"
}
```

Occupancy Sensor

OccSensor.Get

Returns the status of the occupancy sensor.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "OccSensor.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "OccSensor.Get" }
```

Response


```
{
    "id": "OccSensorGetResults",
    "result": {
        "OccupancySensorSettings": {
            "OccupancySensor/enable/on": true,
            "OccupancySensor/enable/off": false,
            "OccupancySensorIP/enable/on": true,
            "OccupancySensorIP/enable/off": false
        },
        "OccupancySensor/OCS-900N-NUM": "2",
        "OccupancySensor/OCS-900N": [
            {
                "ID": "OCS4567",
                "host": "192.168.50.110",
                "port": 9000,
                "connected": true,
                "state": "vacant"
            },
            {
                "ID": "OCS4567",
                "host": "192.168.50.110",
                "port": 9000,
                "connected": true,
                "state": "vacant"
            }
        ]
    },
    "jsonrpc": "2.0"
}
```

OccSensorAdd

Adds the JAV-OCS-900N as a sensor. A maximum of two JAV-OCS-900N sensors can be added.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "OccSensorAdd","params":{"host":"Y","port":Z}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	IP address or hostname
Z	params	Port

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "OccSensorAdd", "params": { "host": "192.168.50.110", "port": 9000 } }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

OccSensorDelete

Removes the specified JAV-OCS-900N sensor. Execute OccSensor.Get to obtain the sensor ID.

Syntax

```
{"jsonrpc":"2.0","id": "X","method": "OccSensorDelete","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	Sensor ID

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "OccSensorDelete", "params": "OCS4567" }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

OccSensorIdentify

Enabled the blinking of the LED indicators on the JAV-OCS-900N to physically identify the unit. The sensor ID must be specified. Use OccSensor.Get to obtain the ID.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "OccSensorIdentify","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	Sensor ID

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "OccSensorIdentify", "params": "OCS4567" }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

OccSensorSettings

Enables or disables the standard I/O and Atlona IP sensors.

Syntax

{"jsonrpc": "2.0","id": "W","method": "OccSensorSettings","params":{"OccupancySensor/enable/on":X,"OccupancySensor/enable/off":X,"OccupancySensorIP/enable/ on":X,"OccupancySensorIP/enable/off":X}}

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

{ "jsonrpc": "2.0", "id": "User1", "method": "OccSensorSettings", "params": { "OccupancySensor/enable/on": true, "OccupancySensor/enable/off": false, "OccupancySensorIP/enable/on": true, "OccupancySensorIP/enable/off": false } }

Response


```
{
    "id": "User1",
    "jsonrpc": "2.0",
    "result": {
        "OccupancySensor/enable/off": false,
        "OccupancySensor/enable/on": false,
        "OccupancySensorIP/enable/off": false,
        "OccupancySensorIP/enable/on": false
    }
}
```

PA Sense

PASense.Get

Returns the PA sense system release delay.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "PASense.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "PASense.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "PASenseStatus": "Connected",
        "PASenseTriggerPort": "Connected",
        "PASenserelease": 15,
        "PASenseTriggerMode": "NO"
    },
    "jsonrpc": "2.0"
}
```

PASenseRelease.Set

Sets PA sense system release delay, in seconds.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "PASenseRelease.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	3 ... 25

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "PASenseRelease.Set", "params": 15 }
```

Response

```
{
    "id": "User1",
    "result": {
        "PASenserelease": 15
    },
    "jsonrpc": "2.0"
}
```

PASenseTriggerMode.Set

Sets the trigger port for PA sense system. The PA sense system uses a contact closure trigger port that can be interfaced to IP PA receivers that have contact closure outputs. When configured as Normally Open (NO), the trigger port is considered inactive when the two pins are not shorted together. When configured as Normally Closed (NC), the pins are shorted together.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "PASenseTriggerMode.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	NO, NC

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "PASenseTriggerMode.Set", "params": "NO" }
```

Response

```
{
    "id": "User1",
    "result": {
        "PASenseTriggerMode": "NO"
    },
    "jsonrpc": "2.0"
}
```

System

Platform.FactoryReset

Resets the unit to factory default settings.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "Platform.FactoryReset"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Platform.FactoryReset" }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

Platform.Reboot

Reboots the unit.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "Platform.Reboot"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Platform.Reboot" }
```

Response

```
{
    "id": "User1",
    "result": true,
    "jsonrpc": "2.0"
}
```

System.Get

Returns information about the unit.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "System.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "System.Get" }
```

Response


```
{
    "id": "User1",
    "result": {
        "HwVer": "v1.0",
        "FwVer": "v1.0",
        "MCUVer": "v1.0",
        "CPLDVer": "v1.0",
        "DSPVer": "v1.0",
        "ValensVer": "v1.0",
        "USB-C": "v1.0",
        "OnTime": "0-19:8:55",
        "model": "AT-OME-CS31-SA-HDBT",
        "serialnumber": "123456",
        "standby": false,
        "frontpanel": true,
        "blinkled": true,
        "network": {
            "hostname": "nil",
            "macaddress": "B8:98:B0:05:93:E6",
            "ipaddress": "192.168.50.134"
        }
    },
    "jsonrpc": "2.0"
}
```

SystemBlinkLed.Set

Enables or disables blinking of the LED indicators on the front panel.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "SystemBlinkLed.Set","params":Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "SystemBlinkLed.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "blinkled": true
    },
    "jsonrpc": "2.0"
}
```

SystemFrontPanel.Set

Enables or disables the front panel buttons.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "SystemFrontPanel.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "SystemFrontPanel.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "frontpanel": true
    },
    "jsonrpc": "2.0"
}
```

SystemStandby.Set

Places the unit in standby mode.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "SystemStandby.Set","params":Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "SystemStandby.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "standby": true
    },
    "jsonrpc": "2.0"
}
```

USB

UsbFollowVideoCfg.Set

Assigns the USB host port to video ports. When the video auto-switches, the USB host port will follow video switching.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "UsbFollowVideoCfg.Set","params":{"source":"Y", "maptohost":"Z"}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3
Z	params	host1, host2

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "UsbFollowVideoCfg. Set", "params": { "source": "in1",
"maptohost": "host1" } }
```

Response

```
{
    "id": "User1",
    "result": {
        "source": "in1",
        "maptohost": "host1"
    },
    "jsonrpc": "2.0"
}
```

UsbSwitch.Get

Returns the current USB switch setting.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "UsbSwitch.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "UsbSwitch.Get" }
```

Response


```
{
    "id": "User1",
    "result": {
        "usbswitchmode": "follow video",
        "followvideousbmap": [
            {
                "source": "in1",
                "maptohost": "host1"
            },
            {
                "source": "in2",
                "maptohost": "host1"
            },
            {
                "source": "in3",
                "maptohost": "host1"
            }
        ],
        "UsbSwitchHost.": "host1",
        "usbVbus": "2"
    },
    "jsonrpc": "2.0"
}
```

UsbSwitchHost.Set

Sets which USB Host is switched to the USB device in manual switching mode.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "UsbSwitchHost.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	host1, host2

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "UsbSwitchHost.Set", "params": "host1" }
```

Response

```
{
    "id": "User1",
    "result": {
        "toHost": "host1"
    },
    "jsonrpc": "2.0"
}
```

UsbSwitchMode.Set

Sets the USB switching mode.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "UsbSwitchMode.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	follow video, manual

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "UsbSwitchMode.Set", "params": "follow video" }
```

Response

```
{
    "id": "User1",
    "result": {
        "usbswitchmode": "follow video"
    },
    "jsonrpc": "2.0"
}
```

UsbVbus.Set

Toggles the Vbus. The default setting is true, which allows the USB hub port to toggle on and off based on the presence of a USB host. If set to false, then USB power to the HOST port is always on. This would allow a device, such as the AT-CAP-SP100, to continually charge, even if no USB host is present.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "UsbVbus.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "UsbVbus.Set", "params": "true" }
```

Response

```
{
    "id": "User1",
    "result": {
        "usbVbus": "true"
    },
    "jsonrpc": "2.0"
}
```

Video

SelectVideoStream.Set

Selects the LumaStream stream the switcher decodes, by stream name (selecttype 0) or by multicast IP address and port (selecttype 1). The response reports the stream now selected, with its name and multicast address. When a stream is selected by multicast address, the response returns an empty stream name until the switcher has resolved the name of the stream at that address. Requires firmware 1.0.2 or later.

Syntax

```
{"jsonrpc": "2.0", "id": "X", "method": "SelectVideoStream.Set", "params": {"selecttype": Y, "streamname": "Z"}}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	selecttype	0: select by stream name 1: select by multicast IP address and port (replace streamname with multiaddr: {"ipaddr": "<address>", "portnumber": <port>})
Z	streamname	Stream name, as returned by VideoStreamNameList.Get

Example

```
{ "jsonrpc": "2.0", "id": "SelectVideoStreamSetResults", "method": "SelectVideoStream.Set", "params": { "selecttype": 0, "streamname": "minions" } }
```

Response

```
{
    "jsonrpc": "2.0",
    "id": "SelectVideoStreamSetResults",
    "result": {
        "multiaddr": {
            "ipaddr": "239.70.2.24",
            "portnumber": 1000
        },
        "selecttype": 0,
        "streamname": "minions"
    }
}
```

VideoAutoSw.Set

Enables or disables auto switching. When set to true, auto switch will be enabled.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoAutoSw.Set","params": Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoAutoSw.Set", "params": true }
```

Response

```
{
    "id": "User1",
    "result": {
        "autosw": true
    },
    "jsonrpc": "2.0"
}
```

VideoHDBTStatus.Get

Returns the HDBaseT status.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoHDBTStatus.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{"jsonrpc": "2.0","id": "User1","method": "VideoHDBTStatus.Get”}
```

Response

```
{
    "id": "User1",
    "result": {
        "link": "up",
        "remote model": "AT-OME-EX-TX"
    },
    "jsonrpc": "2.0"
}
```

VideoHDMIOut5V.Get

Returns the status of HDMI +5 V output when no output signal is present.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoHDMIOut5V.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoHDMIOut5V.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "HDMIOut5V": true
    },
    "jsonrpc": "2.0"
}
```

VideoHDMIOut5V.Set

Enables or disables the HDMI +5 V when no output signal is present.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoHDMIOut5V.Set","params":Y}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoHDMIOut5V.Set", "params": false }
```

Response

```
{
    "id": "User1",
    "result": {
        "HDMIOut5V": false
    },
    "jsonrpc": "2.0"
}
```

VideoInputStatus.Get

Returns the status of the inputs.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoInputStatus.Get","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3, in4 (in4 = Stream input, firmware 1.0.2 or later)

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoInputStatus.Get", "params": "in2" }
```

Response

```
{
    "id": "User1",
    "result": {
        "source": "in2",
        "cableconnection": "Conneted",
        "signal": "Valid",
        "videoformat": "3840x2160,60;none hdr;rgb;8bit;16:9",
        "hdcp": "hdcp2.2",
        "signaltype": "hdmi"
    },
    "jsonrpc": "2.0"
}
```

VideoOutputStatus.Get

Returns the status of the output.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoOutputStatus.Get","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	out1

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoOutputStatus.Get", "params": "out1" }
```

Response

```
{
    "id": "User1",
    "result": {
        "sink": "out1",
        "activeinput": "in1",
        "cableconnection": "Conneted",
        "signal": "Valid",
        "videoformat": "3840x2160,60;none hdr;rgb;8bit;16:9",
        "hdcp": "hdcp2.2",
        "signaltype": "hdmi"
    },
    "jsonrpc": "2.0"
}
```

VideoStreamNameList.Get

Returns the LumaStream video streams currently available to the switcher. The names returned are the stream names published by the LumaStream encoders on the network, and are the values accepted by SelectVideoStream.Set. Requires firmware 1.0.2 or later.

Syntax

```
{"jsonrpc": "2.0", "id": "X", "method": "VideoStreamNameList.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.

Example

```
{ "jsonrpc": "2.0", "id": "VideoStreamNameListGetResults", "method": "VideoStreamNameList.Get" }
```

Response

```
(no feedback)
```

VideoSwitch.Get

Returns the current switching settings.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoSwitch.Get"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	out1

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoSwitch.Get" }
```

Response

```
{
    "id": "User1",
    "result": {
        "autoswitch": true,
        "switch": {
            "in": "in1",
            "out": "out1"
        },
        "switchdefault": "in1",
        "jsonrpc": "2.0"
    }
}
```

VideoSwitch.Set

Switches the specified input to the output.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoSwitch.Set"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3, in4 (in4 = Stream input, firmware 1.0.2 or later)
Y	params	out1

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoSwitch.Set", "params": { "in": "in1", "out": "out1" } }
```

Response

```
{
    "id": "User1",
    "result": {
        "switch": {
            "in": "in1",
            "out": "out1"
        }
    },
    "jsonrpc": "2.0"
}
```

VideoSwitchDefault.Set

Assigns the default input to be switched to, when no video is present on the last connected input, in auto switching mode.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoSwitchDefault.Set","params": "Y"}
```

Identifier	Request Object	Parameters
X	id	Enter a user id. This is optional.
Y	params	in1, in2, in3, in4 (in4 = Stream input, firmware 1.0.2 or later)

Example

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoSwitchDefault.Set", "params": "in1" }
```

Response

```
{
    "id": "User1",
    "result": {
        "switchdefault": "in1"
    },
    "jsonrpc": "2.0"
}
```