AT-OME-SW21-TX

Omega™ 2x1 AV Switcher HDBaseT Transmitter w/USB Hub

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
AV....5
HDCP....5
EDID....5
USB....6
Display....6
Occupany Sensor....6
Network....7
System....7
Commands....8
AV....8
HDCP....22
EDID....24
USB....26
Display....30
Occupany Sensor....41
Network....46
System....52

Version Information

This document lists the commands for controlling the AT-OME-SW21-TX over Telnet, SSH, WebSocket (JSON-RPC), and RS-232. Its version history follows the original command reference.

Version	Date	Notes
1	May 2025	Factory release
HR 1.0	October 2026	Converted to the Hall Research document format. No command changes.

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-SW21-TX)	https://www.hallresearch.com/product/AT-OME-SW21-TX

IMPORTANT: Visit hallresearch.com/product/AT-OME-SW21-TX for the latest firmware updates and documentation.

Introduction

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	HDBaseT OUT
9002	RS-232 port

NOTE: This product does not support TCP Proxy port 9000.

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; do not change spacing or lettering. Each command is terminated with a carriage return. If a command fails or is entered incorrectly, the feedback is “Unknown command”.

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.

AV

Command	Description
VideoSwitch.Get	Displays the current I/O settings
VideoAutoSw.Set	Sets auto-switching to on/off
VideoSwitch.Set	Switches input to output
VideoInputStatus.Get	Gets the status of the input
AudioInputFormat.Get	Gets input audio format information
VideoOutputStatus.Get	Gets the status of the output
AudioOutFormat.Get	Gets the audio format information of output
VideoHDMIOut5V.Set	Enables/disables +5V when no output signal is present
VideoHDMIOut5V.Get	Gets the status of HDMI output +5V when no output signal is present
VideoHDBTStatus.Get	Gets the HDBT status
AudioOutMute.Set	Sets the audio out's mute status
AudioOutMute.Get	Gets the audio out's mute status
VideoScaler.Set	Enables/disables the scaler on the HDMI output
VideoScaler.Get	Gets the status of the scaler on the HDMI output

HDCP

Command	Description
HdcpCompliant.Set	Sets HDCP compliance for the inputs
HdcpCompliant.Get	Gets the HDCP status of the inputs

EDID

Command	Description
EdidInput.Set	Sets input EDID
EDIDInput.Get	Gets the inputs' EDID status

USB

Command	Description
UsbSwitch.Get	Displays the current USB switching settings
UsbSwitchMode.Set	Sets USB Host switching mode
UsbSwitchHost.Set	Sets which USB Host is switched to USB device on manual mode
UsbVbus.Set	To "bounce" Vbus on USB device hub when host is switched

Display

Command	Description
DisplayCtrl.Get	Gets the display control mode
DisplayCtrlMode.Set	Sets the control method of the display to CEC, RS-232 or IP
DisplayCtrlOperation.Set	Controls the Display
DisplayPowerOnAuto.Set	Enable to set the display to power off when the power settings are met
DisplayCtrlDelay.Set	Sets the delay for the unit to control the display power on/off
DisplayDisplayWarm.Set	Sets the time between when the projector lamp has been turned on to when it can receive new commands
DisplayCtrlLampCoolDown.Set	Sets the time between when projector lamp has been turned off to when it can receive new commands
DisplayCtrlIP.Set	Sets the IP parameters for the unit to send commands to the display
DisplayCtrlRs232.Set	Sets the RS-232 parameters for the unit to send commands to the display
DisplayCtrlCmd.Set	Sets the command string to send to the display

Occupany Sensor

Command	Description
OccSensor.Get	Gets the status of the Occupancy Sensor
OccSensorAdd	Adds the JAV-OCS-900N as a sensor
OccSensorDelete	Removes the JAV-OCS-900N as a sensor
OccSensorIdentify	The LEDs of the JAV-OCS-900N will blink to help identify a specific unit when there are more than one
OccSensorSettings	Enables/Disables the JAV-OCS-900N will send auto on/off commands

Network

Command	Description
Network.Get	Gets the current network settings
Network.Set	Sets the network settings for the switcher
NetworkHostname.Set	Sets the unit's host name
NetworkHostname.Get	Gets the unit's host name
NetworkCtlProtocolsEnable.Set	Sets the network control protocol was enabled or disabled
NetworkCtlProtocols.Get	Displays the network control protocol

System

Command	Description
Platform.FactoryReset	Factory resets the unit
Platform.Reboot	Reboots the unit
System.Get	Displays the current system's information
SystemFrontPanel.Set	Enables/disables the front panel buttons to help installer identify the device
SystemHdbt.Get	Get the System HDBaseT Info
SystemHdbtStart.Set	Set the System HDBaseT information to start or stop

Commands

AV

VideoSwitch.Get

Displays the current I/O settings.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoSwitchGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

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

VideoAutoSw.Set

Sets auto-switching to on/off.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoSwitchGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false true: Auto-switching enabled. false: Auto-switching disabled.

Example

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

Response

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

VideoSwitch.Set

Switches input to output.

Syntax

```
{"jsonrpc": "2.0","id": "X","method": "VideoSwitch.Set","params":{"in": "Y","out": "Z"}}
```

Identifier	Request Object	Parameters
X	id	Default: VideoSwitchSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	in	in1, in2 in1: USB-C IN in2: HDMI IN
Z	out	out1, out2 out1: HDMI OUT out2: HDBaseT OUT

Example

```
{ "jsonrpc": "2.0", "id": "VideoSwitchSetResults", "method": "VideoSwitch.Set", "params": { "in": "in2", "out": "out1" } }
```

Response

```
{
    "id": "VideoSwitchSetResults",
    "jsonrpc": "2.0",
    "result": {
        "in": "in2",
        "out": "out1"
    }
}
```

VideoInputStatus.Get

Gets the status of the input.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoInputStatusGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	in1, in2 in1: USB-C IN in2: HDMI IN

Example

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

Response

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

AudioInputFormat.Get

Gets input audio format information.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: AudioInputFormatGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	in1, in2 in1: USB-C IN in2: HDMI IN

Example

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

Response

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

VideoOutputStatus.Get

Gets the status of the output.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoOutputStatusGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	out1, out2 out1: HDMI OUT out2: HDBaseT OUT

Example

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

Response

```
{
    "id": "VideoOutputStatusGetResults",
    "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"
}
```

AudioOutFormat.Get

Gets the audio format information of output.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoOutputStatusGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	out1, out2 out1: HDMI OUT out2: HDBaseT OUT

Example

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

Response

```
{
    "id": "AudioOutFormatGetResults",
    "jsonrpc": "2.0",
    "result": {
        "audioformat": "48kHZ",
        "sink": "out1"
    }
}
```

VideoHDMIOut5V.Set

Enables/disables +5V when no output signal is present.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoHDMIOut5VSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "VideoHDMIOut5VSetResults", "method": "VideoHDMIOut5V.Set", "params": false }
```

Response

```
{
    "id": "VideoHDMIOut5VSetResults",
    "result": {
        "HDMIOut5V": false
    }
    "jsonrpc": "2.0"
}
```

VideoHDMIOut5V.Get

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

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoHDMIOut5VSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

```
{ "jsonrpc": "2.0", "id": "VideoHDMIOut5VGetResults", "method": "VideoHDMIOut5V.Get" }
```

Response

```
{
    "id": "VideoHDMIOut5VGetResults",
    "result": {
        "HDMIOut5V": true
    }
    "jsonrpc": "2.0"
}
```

VideoHDBTStatus.Get

Gets the HDBT status.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoHDMIOut5VSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
"remote model": "AT-OME-EX-RX"
{ }
"id": "X", }
"jsonrpc": "2.0",
"result": {
    "link": "up",
    "remote ip": "",
```

AudioOutMute.Set

Sets the audio out's mute status.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: AudioOutMuteSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "AudioOutMuteSetResults", "method": "AudioOutMute.Set", "params": false }
```

Response

```
{
    "id": "AudioOutMuteSetResults",
    "result": {
        "AudioOutMute": false
    }
    "jsonrpc": "2.0"
}
```

AudioOutMute.Get

Gets the audio out's mute status.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: AudioOutMuteGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

```
{ "jsonrpc": "2.0", "id": "AudioOutMuteGetResults", "method": "AudioOutMute.Get" }
```

Response

```
{
    "id": "AudioOutMuteGetResults",
    "result": {
        "AudioOutMute": false
    }
    "jsonrpc": "2.0"
```

VideoScaler.Set

Enables/disables the scaler on the HDMI output.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoScalerSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "VideoScalerSetResults", "method": "VideoScaler.Set", "params": true }
```

Response

```
{
    "id": "VideoScalerSetResults",
    "result": {
        "Scaler": true
    }
    "jsonrpc": "2.0"
}
```

VideoScaler.Get

Gets the status of the scaler on the HDMI output.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: VideoScalerGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

```
{ "jsonrpc": "2.0", "id": "VideoScalerGetResults", "method": "VideoScaler.Get" }
```

Response

```
{
    "id": "VideoScalerGetResults",
    "result": {
        "Scaler": true
    }
    "jsonrpc": "2.0"
```

HDCP

HdcpCompliant.Set

Sets HDCP compliance for the inputs.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: HdcpCompliantSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	source	in1, in2 in1: USB-C IN in2: HDMI IN
S	hdcpCompliant	true, false

Example

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

Response

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

HdcpCompliant.Get

Gets the HDCP status of the inputs.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: HdcpCompliantGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

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

EDID

EdidInput.Set

Sets input EDID.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: EdidInputSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	source	in1, in2 in1: USB-C IN in2: HDMI IN
Z	edidmode	1-10, 98, 99 1: Copy from the hdmi output connected display (Default) 2: Fixed 3840x2160@60 Hz 2.0CH PCM Audio with HDR; 3: Fixed 3840x2160@30 Hz 2.0CH PCM Audio with SDR; 4: Fixed 1920x1080P@60Hz 2.0CH PCM Audio with SDR; 5: Fixed 1920x1080P@30Hz 2.0CH PCM Audio with SDR; 6: Fixed 1280x720P@60Hz 2.0CH PCM Audio with SDR; 7: Fixed 1280x800 2.0CH PCM Audio with SDR; 8: Fixed 1920x1200 2.0CH PCM Audio with SDR; 9: Fixed 3440x1440@50 2.0CH PCM Audio with SDR 10: Fixed 2560x1080@60 2.0CH PCM Audio with SDR 99: mem1 98: mem2

Example

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

Response

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

EDIDInput.Get

Gets the inputs' EDID status.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: EDIDInputGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "EDIDInputGetResults",
    "result": {
        "mem1": true,
        "mem1EDID ": "xxxx",
        "mem2": true,
        "mem2EDID ": "xxxx",
        "edidsettings": [
            {
                "source": "in1",
                "edidmode": 2
            },
            {
                "source": "in2",
                "edidmode": 2
            }
        ]
    },
    "jsonrpc": "2.0"
}
```

USB

UsbSwitch.Get

Displays the current USB switching settings.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: UsbSwitchGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "UsbSwitchGetResults",
    "jsonrpc": "2.0",
    "result": {
        "followvideousbmap": [
            {
                "maptohost": "host1",
                "source": "in1"
            },
            {
                "maptohost": "host2",
                "source": "in2"
            }
        ],
        "usbSwitchHost": "host1",
        "usbVbus": "true",
        "usbswitchmode": "follow video"
    }
}
```

UsbSwitchMode.Set

Sets USB Host switching mode. Default is follow video.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: UsbSwitchModeSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	follow video, manual

Example

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

Response

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

UsbSwitchHost.Set

Sets which USB Host is switched to USB device on manual mode.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: UsbSwitchHostSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	host1, host2, host3

Example

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

Response

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

UsbVbus.Set

To "bounce" Vbus on USB device hub when host is switched. Default is set to true.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: UsbVbusSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

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

Response

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

Display

DisplayCtrl.Get

Gets the display control mode.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
x40\\x04",
{ "DisplayControl/command/unmute":
    "id": "X", "\\x50\\x44\\x43",
    "jsonrpc": "2.0", "DisplayControl/command/
    "result": { volumedown": "\\x50\\x44\\x42",
        "DisplayControl/AutocontrolDelay": "DisplayControl/command/volumeup":
        120, "\\x50\\x44\\x41",
        "DisplayControl/ "DisplayControl/controlmethod":
        AutocontrolEnable": "VideoPresence", "CecAPI",
        "DisplayControl/DisplayWarmTime": "DisplayControl/ip/address": "",
        "10", "DisplayControl/ip/mode": "Non-
        "DisplayControl/LampCoolDownTime": Login",
        "10", "DisplayControl/ip/password": "",
        "DisplayControl/RS232": [ "DisplayControl/ip/port": "",
            { "DisplayControl/ip/username": "",
                "Baudrate": "115200", "DisplayControl/manufacturer": "",
                "Databits": "8", "DisplayControl/model": "",
                "Parity": "N", "DisplayControl/products": ""
                "Stopbits": "1", }
            "port": "uart1" }
    },
    {
        "Baudrate": "115200",
        "Databits": "8",
        "Parity": "N",
        "Stopbits": "1",
        "port": "uart2"
    }
],
"DisplayControl/command/
RepeatStatus": false,
"DisplayControl/command/
RepeatTimes": "2",
"DisplayControl/command/mute": "\\
x50\\x44\\x43",
"DisplayControl/command/off": "\\
x40\\x36",
"DisplayControl/command/on": "\\
```

DisplayCtrlMode.Set

Sets the control method of the display to CEC, RS-232 or IP.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlModeSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	IP, RS232, CecAPI

Example

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

Response

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

DisplayCtrlOperation.Set

Controls the Display.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlOperationSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	source	poweron, poweroff, volumeup, volumedown, mute, unmute
Z	controlmode	IP, RS232, CecAPI

Example

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

Response

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

DisplayPowerOnAuto.Set

Enable to set the display to power off when the power settings are met. The display will automatically turn on as soon as a signal is received and all timers have expired. Modes Description Active Video Presence Device will send the power off command to the display if no active source is detected on the input, and power on command when an active source is detected. Power timers will be followed. Active Video Presence When the occupancy sensor (JAV-OCS-900N) is triggered and source signal is w/Occupancy Sensor* active or inactive, it will send the on or off command based on physical and signal presence. Occupancy Sensor only Power on and off commands will be sent based on the JAV-OCS-900N sensor status. The sensor must be connected to the same network as the OME-SW21- TX and connected on the I/O page. Disabled No display control

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlModeSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	VideoPresence, Video/OccSensor, OccSensor, Disabled

Example

```
{ "jsonrpc": "2.0", "id": "DisplayPowerOnAutoSetResults", "method": "DisplayPowerOnAuto.Set", "params": "Video/OccSensor" }
```

Response

```
{
    "id": "DisplayPowerOnAutoSetResults",
    "jsonrpc": "2.0",
    "result": {
        "displaypowonauto": "Video/OccSensor"
    }
}
```

DisplayCtrlDelay.Set

Sets the delay for the unit to control the display power on/off. Parameter is set in seconds.

NOTE: The parameter will not be triggered unless DisplayPowerOnAuto.Set is set to true.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlDelaySetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	15-900

Example

```
{ "jsonrpc": "2.0", "id": "DisplayCtrlDelaySetResults", "method": "DisplayCtrlDelay.Set", "params": 120 }
```

Response

```
{
    "id": "DisplayCtrlDelaySetResults",
    "jsonrpc": "2.0",
    "result": {
        "displayctrldelay": 120
    }
}
```

DisplayDisplayWarm.Set

Sets the time between when the projector lamp has been turned on to when it can receive new commands.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayDisplayWarmSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	10-300

Example

```
{ "jsonrpc": "2.0", "id": "DisplayDisplayWarmSetResults", "method": "DisplayDisplayWarm.Set", "params": "15" }
```

Response

```
{
    "id": "DisplayDisplayWarmSetResults",
    "jsonrpc": "2.0",
    "result": {
        "DisplayWarmTime": "15"
    }
}
```

DisplayCtrlLampCoolDown.Set

Sets the time between when projector lamp has been turned off to when it can receive new commands. Parameter is set in seconds.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlLampCoolDownSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	10-300

Example

```
{ "jsonrpc": "2.0", "id": "DisplayCtrlLampCoolDownSetResults", "method":
"DisplayCtrlLampCoolDown.Set", "params": "15" }
```

Response

```
{
    "id": "DisplayCtrlLampCoolDownSetResults",
    "jsonrpc": "2.0",
    "result": {
        "LampCoolDownTime": "15"
    }
}
```

DisplayCtrlIP.Set

Sets the IP parameters for the unit to send commands to the display.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlIPSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	ipaddress	xxx.xxx.xxx.xxx
Z	port	xxx (up to a 5 digit port)
M	mode	Login, Non-Login
N	username /password	xx (username and password can be up to 64 characters)

Example

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

Response

```
{
    "id": "DisplayCtrlIPSetResults",
    "jsonrpc": "2.0",
    "result": {
        "ipaddress": "192.168.23.189",
        "mode": "Login",
        "password": "admin",
        "port": 23,
        "username": "admin"
    }
}
```

DisplayCtrlRs232.Set

Sets the RS-232 parameters for the unit to send commands to the display.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: DisplayCtrlRs232SetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
W	port	uart1, uart2
Y	baudrate	9600, 19200, 38400, 57600, 115200
Z	parity	N (none), O (odd), E (even)
M	dataBit	7, 8
N	stopBit	1, 2

Example

```
{ "jsonrpc": "2.0", "id": "DisplayCtrlRs232SetResults", "method": "DisplayCtrlRs232.Set", "params": { "port": "uart1", "baudrate": "115200", "parity": "N", "dataBit": "8", "stopBit": "1" } }
```

Response

```
{
    "id": "DisplayCtrlRs232SetResults",
    "result": true,
    "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	Default: DisplayCtrlCmdSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	cmd	poweron, poweroff, volumeup, volumedown, mute, unmute
Z	data	xxxxxxxxxx Command for display control

Example

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

Response

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

Occupany Sensor

OccSensor.Get

Gets the status of the Occupancy Sensor.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: OccSensorGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "OccSensorGetResults",
    "result": {
        "OccupancySensorSettings":
        {
            "OccupancySensorIP/enable/off": true,
            "OccupancySensorIP/enable/on": true
        },
        "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. Up to 2 JAV-OCS-900Ns can be added.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: OccSensorAddResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	host	IP address or hostname
Z	port	Port

Example

```
{ "jsonrpc": "2.0", "id": "OccSensorAddResults", "method": "OccSensorAdd", "params": { "host": "192.168.23.218", "port": 9000 } }
```

Response

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

OccSensorDelete

Removes the JAV-OCS-900N as a sensor.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: OccSensorDeleteResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	OCS Sensor ID

Example

```
{ "jsonrpc": "2.0", "id": "OccSensorDeleteResults", "method": "OccSensorDelete", "params": "OCS1234" }
```

Response

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

OccSensorIdentify

The LEDs of the JAV-OCS-900N will blink to help identify a specific unit when there are more than one.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: OccSensorIdentifyResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	OCS Sensor ID

Example

```
{ "jsonrpc": "2.0", "id": "OccSensorIdentifyResults", "method": "OccSensorIdentify", "params":
"OCS1234" }
```

Response

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

OccSensorSettings

Enables/Disables the JAV-OCS-900N will send auto on/off commands.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: OccSensorSettingsResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

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

Response

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

Network

Network.Get

Gets the current network settings.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "NetworkGetResults",
    "jsonrpc": "2.0",
    "result": {
        "gateway": "192.168.23.1",
        "ip_mode": "dhcp",
        "ip_port": 23,
        "ipaddr": "192.168.23.214",
        "mac": "34:1b:22:80:11:44",
        "netmask": "255.255.255.0"
    }
}
```

Network.Set

Sets the network settings for the switcher.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
V	ip_mode	static, dhcp DHCP is selected by default
Y	ipaddr	IP address (e.g. 169.254.9.9)
Z	netmask	Netmask (e.g. 255.255.0.0)
I	gateway	Gateway (e.g. 192.168.2.254)
W	ip port	Port number

Example

```
{ "jsonrpc": "2.0", "id": "NetworkSetResults", "method": "Network.Set", "params": { "ip_mode": "static",
"ipaddr": "192.168.23.214", "netmask": "255.255.255.0", "gateway": "192.168.23.1", "ip_port": 23 } }
```

Response

```
{
    "id": "NetworkSetResults",
    "jsonrpc": "2.0",
    "result": {
        "gateway": "192.168.23.1",
        "ip_mode": "static",
        "ip_port": 23,
        "ipaddr": "192.168.23.214",
        "netmask": "255.255.255.0"
    }
}
```

NetworkHostname.Set

Sets the unit's host name.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkHostnameSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	xxx (user created host name), nil Default is nil

Example

```
{ "jsonrpc": "2.0", "id": "NetworkHostnameSetResults", "method": "NetworkHostname.Set", "params": "AT-OME-SW21TX" }
```

Response

```
{
    "id": "NetworkHostnameSetResults",
    "jsonrpc": "2.0",
    "result": {
        "hostname": "AT-OME-SW21TX"
    }
}
```

NetworkHostname.Get

Gets the unit's host name.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkHostnameGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "NetworkHostnameGetResults",
    "jsonrpc": "2.0",
    "result": {
        "hostname": "AT-OME-SW21TX-1"
    }
}
```

NetworkCtlProtocolsEnable.Set

Sets the network control protocol was enabled or disabled.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkCtlProtocolsEnableSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	host	telnet, ssh
Z	port	true, false default is true

Example

```
{ "jsonrpc": "2.0", "id": "NetworkCtlProtocolsEnableSetResults", "method": "NetworkCtlProtocolsEnable.Set", "params": { "protocol": "ssh", "enabled": true } }
```

Response

```
{
    "id": "NetworkCtlProtocolsEnableSetResults",
    "jsonrpc": "2.0",
    "result": {
        "enabled": true,
        "protocol": "ssh"
    }
}
```

NetworkCtlProtocols.Get

Displays the network control protocol.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: NetworkCtlProtocolsGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

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

System

Platform.FactoryReset

Factory resets the unit.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: PlatformFactoryResetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

```
{
    "id": "PlatformFactoryResetResults",
    "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	Default: PlatformRebootResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response

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

System.Get

Displays the current system's information.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: SystemGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

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

Response


```
{
    "id": "SystemGetResults",
    "jsonrpc": "2.0",
    "result": {
        "FwVer": "0.0.0.18",
        "HwVer": "V0.1",
        "OnTime": "0-18:49:26",
        "SerialNumber": "",
        "USB-C": "1.3.1",
        "ValensVer": "7.5.23",
        "blinkled": false,
        "frontpanel": false,
        "model": "AT-OME-SW21TX",
        "network": {
            "hostname": "AT-OME-SW21TX-1",
            "ipaddress": "192.168.23.214",
            "macaddress": "34:1b:22:80:11:44"
        },
        "standby": false
    }
}
```

SystemFrontPanel.Set

Enables/disables the front panel buttons to help installer identify the device.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: SystemFrontPanelSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

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

Response

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

SystemHdbt.Get

Get the System HDBaseT Info.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: SystemHdbtGetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands

Example

```
{ "jsonrpc": "2.0", "id": "SystemHdbtGetResults", "method": "SystemHdbt.Get" }
```

Response

```
{
    "id": "SystemHdbtGetResults",
    "jsonrpc": "2.0",
    "result": {
        "TxVersion": "V1.0.0",
        "RxVersion": "V1.0.0",
        "Tmdsclk": "148 Mhz",
        "CableLength": "20 meters",
        "VideoQ": "Pass",
        "QualityA": "Pass",
        "QualityB": "Pass",
        "QualityC": "Pass",
        "QualityD": "Pass",
    }
}
```

SystemHdbtStart.Set

Set the System HDBaseT information to start or stop.

Syntax

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

Identifier	Request Object	Parameters
X	id	Default: SystemHdbtStartSetResults *Optional* Type in a user id, this will return with each feedback, to provide a way to see the difference between commands
Y	params	true, false

Example

```
{ "jsonrpc": "2.0", "id": "SystemHdbtStartSetResults", "method": "SystemHdbtStart.Set", "params": true }
```

Response

```
{
    "id": "SystemHdbtStartSetResults",
    "result": {
        "Start": true,
    },
    "jsonrpc": "2.0"
}
Toll free US International
```