ATLONA
a PANDUIT company

OMEGA™

4K / UHD
5x2 Matrix Switcher
for HDMI, USB-C, DisplayPort, and HDBaseT™ with USB and Wireless Link

Application Programming Interface
2.9.4

AT-OME-MS52W

Atlona Manuals
Switchers

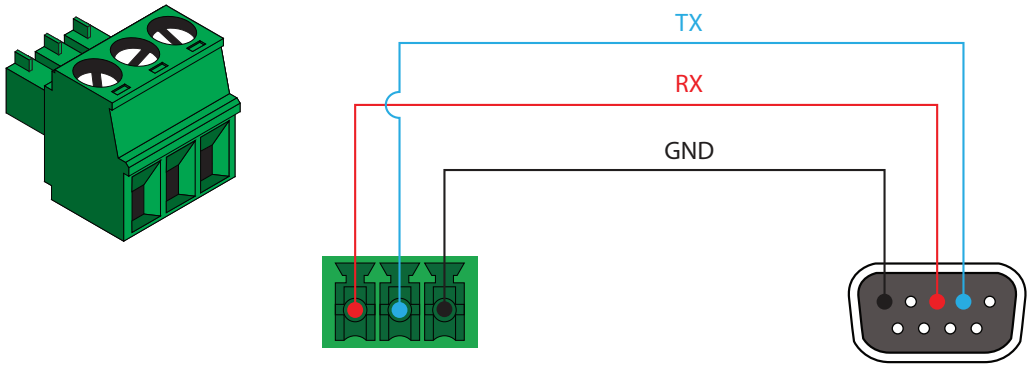
Version Information

Version	Release Date	Notes
1	Jul 2020	Initial release
2	Nov 2020	**Firmware 2.9.0** - no API changes.
3	Feb 2021	**Firmware 2.9.1** - Added following commands: Input:Get Input:Set SetVol
4	Mar 2022	**Firmware 2.9.4** - no API changes

Introduction

RS-232

A 3-pin captive screw connector has been included for RS-232.



RS-232 is often connected through a DB 9-pin to captive screw connector. Each pin is assigned a specific function. Not all pins will be used.

The pin-out will be determined by the RS-232 cable and connected as RX (receive), TX (transmit) and GND (ground).

NOTE: Typical DB9 connectors use pin 2 for TX, pin 3 for RX, and pin 5 for ground. On some devices functions of pins 2 and 3 are reversed.

Baud parameters must be met for control signals to pass. The parameters can be updated through the built-in web server. The defaults for the RS-232 ports are: 9600, N, 8, 1.

Commands

The following is a list of Telnet commands for the AT-OME-MS52W. Commands are *not* case-sensitive. Do not change spacing or lettering. Each command is terminated with a carriage return. If the command fails or is entered incorrectly, then the feedback is “Unknown command”.

IMPORTANT: Each command must be terminated with a carriage-return (0x0d) and the feedback is terminated with a carriage-return and line-feed (0x0a). In addition, when sending multiple commands, at least 500 milliseconds should be specified between each command.

Telnet Commands

Command	Description
?	Lists all available commands
Audio:GetBass	Returns the current bass setting
Audio:GetSource	Returns the current audio source type
Audio:GetTreble	Returns the current treble setting
Audio:Mute:Get	Returns the muting state of the audio output
Audio:Mute:Set	Sets the muting state for HDMI or analog audio output
Audio:SetBass	Sets the amount of bass effect applied to the audio ouptut
Audio:SetSource	Sets the external audio source to analog or digital
Audio:SetTreble	Sets the amount of treble effect applied to the audio ouptut
Audio:Volume:Decrease	Decreases the audio output level by the specified amount
Audio:Volume:Get	Returns the current audio output level
Audio:Volume:Increase	Increases the audio output level by the specified amount
Audio:Volume:Set	Sets the overall audio output level
Calendar:Refresh	Refreshes the calendar
CEC:Trig	Sends the power-on or power-off command to the display using CEC
Display:BYOD:Kick	Removes (“kicks”) the BYOD user from the AT-OME-MS52W
Display:Control:IP:Get	Returns the IP address and port of the remote IP device
Display:Control:IP:Set	Sets the IP address and port of the remote IP device
Display:Input:All:Get	Returns a list of all inputs and the current status
Display:Input:Get	Returns the active input
Display:Input:HDCP:State:Get	Returns the HDCP input status
Display:Input:HDCP:State:Set	Sets the HDCP state of the specified input
Display:Input:Set	Sets the active input
Display:InputState:Get	Returns the input state of the specified input
Display:Matrix:Get	Returns the input for the specified output
Display:Matrix:Mode:Get	Returns the Matrix Mode state
Display:Matrix:Mode:Set	Enables or disables Matrix Mode
Display:Matrix:Set	Routes the specified input to the desired output
Display:Minimal:Get	Gets the state of the output display
Display:Minimal:Set	Sets the state for the output display
GetBroadcast	Returns the current broadcast setting value
GetHostName	Returns the hostname of the AT-OME-MS52W
GetInputStates	Returns the state of each input on the AT-OME-MS52W
GetNoSignalTimeout	Returns the current time interval before the connected display is powered-off
Input:Get	Returns the active input
Input:Set	Sets the active input
Instruments:Temperature:Get	Returns the internal temperature of the AT-OME-MS52W

Commands

Command	Description
Misc:Model:Get	Returns the model of AT-OME-MS52W
Misc:Version:Get	Returns the fimware version of the specified system
Misc:Versions:Get	Returns the firmware version of all systems
Moderator:Enable:Get	Returns the status for Moderator mode
Moderator:Enable:Set	Enable or disable Moderator mode
Moderator:Kick	Kicks the specified BYOD device from the AT-OME-MS52W
Moderator:Show	Sets the specified BYOD device ID for casting
Moderator:Status:Get	Returns the status of Moderator mode
Net:GetInfo	Returns information about the network interface.
OutputMode	Sets the output mode for command feedback
Platform:Reset	Resets the AT-OME-MS52W to factory-default settings
Platform:Restart	Reboots the AT-OME-MS52W
Platform:Shutdown	Shuts down (powers-off) the AT-OME-MS52W
Quit	Ends the Telnet session
Relay:PulseDuration:Get	Returns the relay pulse duration interval
Relay:PulseDuration:Set	Sets the relay pulse duration interval
Relay:State:Get	Returns the state of the relay
Relay:State:Set	Sets the relay state
SetBroadcast	Enables or disables broadcast mode
SetVol	Shorthand command version for controlling audio volume
USBRouting:Follow:Input	Assigns the video input to awith the specified USB host when using Follow USB mode
USBRouting:Input:Get	Returns the USB host and corresponding input for the Manual USB Routing mode
USBRouting:Input:Set	Sets the active USB host port
USBRouting:Mode:Get	Returns the USB routing mode
USBRouting:Mode:Set	Sets the USB routing mode
Zone:PortParams	Returns the settings for the specified zone
Zone:PortSetup	Sets the RS-232 settings for the specified zone
Zone:SendCmd	Sends a command to the specified zone

?

Displays a list of available commands. Note that the example feedback, below, is not a listing of all commands. To display help for a specific command, use the Help command.

Syntax
?

This command does not require any parameters

Example

```
?
```

Feedback

```
Commands Listing

------------------------------------------------------------------
-
                            ? - This list
                         Help - More detailed help on a specific command
                         Quit - End the session
                   OutputMode - Change the output mode for all commands
              Misc:Model:Get - Returns model number of the device
            Misc:Version:Get - Returns version numbers for the various subsystems
           Misc:Versions:Get - Returns version numbers for the various subsystems
            Platform:Restart - Restarts the device
           Platform:Shutdown - Shutdown the device
              Platform:Reset - Factory Resets the device or certain subsystems
        Display:Minimal:Set - Turn the Display on / off [ 0 - 1 ]
        Display:Minimal:Get - Get the Display on / off
      Display:Input:All:Get - Return the active inputs
          Display:Input:Get - Return the active input
          Display:Input:Set - Changes the active input
Display:Input:HDCP:State:Get - Get the state of HDCP on an input  on / off
Display:Input:HDCP:State:Set - Turn HDCP on an input  on / off
          Display:BYOD:Kick - Disconnects all existing BYOD clients
     Display:Control:IP:Get - Return the Display Control IP address and port
     Display:Control:IP:Set - Sets the Display Control IP address and port
             GetInputStates - Return the state of all inputs
...
...
...
...
```

Audio:GetBass

Returns the current bass setting.

Syntax
Audio:GetBass

This command does not require any parameters

Example

```
Audio:GetBass
```

Feedback

```
{
  "result": {
    "bass": {
      "units": "dB",
      "value": 0
    }
  },
  "methodreturn": "Audio: GetBass"
}
```

Audio:GetSource

Returns the current audio source type. The returned value can be either `digital` or `analog`.

Syntax
Audio:GetSource

This command does not require any parameters

Example

```
Audio:GetSource
```

Feedback

```
{
  "result": {
    "audiosource": "digital"
  },
  "methodreturn": "Audio: GetSource"
}
```

Audio:GetTreble

Returns the current treble setting.

Syntax
Audio:GetTreble

This command does not require any parameters

Example

```
Audio:GetTreble
```

Feedback

```
{
  "result": {
    " treble": {
      "units": "dB",
      "value": 0
    }
  },
  "methodreturn": "audio: gettreble"
}
```

Audio:Mute:Get

Returns the muting status.

Syntax
Audio:Mute:Get

This command does not require any parameters

Example

```
Audio:Mute:Get
```

Feedback

```
{
  "
  result": {
    "outputmute": {
      "analog": false,
      "hdmi": true
    }
  },
  "methodreturn": "Audio: Mute: Get"
}
```

Audio:Mute:Set

Sets the muting for the HDMI or analog audio.

Syntax
Audio:Mute:Set X Y

Parameter	Description	Range
X	HDMI audio muting	true, false
Y	Analog audio muting	true, false

Example

```
Audio:Mute:Set hdmi true
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "audio: mute: set hdmi true"
}
```

Audio:SetBass

Sets the amount of bass effect applied to the audio output.

Syntax
Audio:SetBass

Parameter	Description	Range
X	Amount of bass effect (dB)	-12 ... 15

Example

```
Audio:SetBass 2
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "audio: setbass 2"
}
```

Audio:SetSource

Assigns the input audio source.

Syntax
Audio:SetSource

Parameter	Description	Range
X	Audio source	digital, analog

Example

```
Audio:SetSource analog
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "audio: setsource analog"
}
```

Audio:SetTreble

Sets the amount of treble effect applied to the audio output.

Syntax
Audio:SetTreble

Parameter	Description	Range
X	Amount of treble effect (dB)	-12 ... 15

Example

```
Audio:SetTreble 5
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "audio: settreble 5"
}
```

Audio:Volume:Decrease

Decreases the output volume by a specified amount. Values are measured in decibels.

Syntax
Audio:Volume:Decrease X

Parameter	Description	Range
X	Amount to decrease audio (in dB)	1 ... 80

Example

```
Audio:Volume:Decrease 20
```

Feedback

```
{
  “
  result”: {
    “volume”: -20,
    “success”: true
  },
  “methodreturn”: “audio: volume: decrease 20”
}
```

Audio:Volume:Get

Returns the current audio output level in decibels.

Syntax
Audio:Volume:Get

This command does not require any parameters

Example

```
Audio:Volume:Get
```

Feedback

```
Audio: Volume: Get {
  "result": {
    "volume": {
      "units": "dB",
      "value": 0
    }
  },
  "methodreturn": "Audio:Volume:Get"
}
```

Audio:Volume:Increase

Increases the output volume by a specified amount. Values are measured in decibels.

Syntax
Audio:Volume:Increase X

Parameter	Description	Range
X	Amount to increase audio (in dB)	1 ... 80

Example

```
Audio:Volume:Increase 25
```

Feedback

```
{
  "result": {
    "volume": 0,
    "success": true
  },
  "methodreturn": "audio: volume: increase 25"
}
```

Audio:Volume:Set

Sets the overall audio output level in decibels.

Syntax
Audio:Volume:Set X

Parameter	Description	Range
X	Output level (in dB)	-80 ... 0

Example

```
Audio:Volume:Set -10
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "audio: volume: set - 10"
}
```

Calendar:Refresh

Refreshes the calendar.

Syntax
Calendar:Refresh

This command does not require any parameters

Example

```
Calendar:Refresh
```

Feedback

```
{
  "result": {
    "response": ""
  },
  "methodreturn": "calendar: refresh",
  "jsonrpc": "2.0"
}
```

CEC:Trig

Sends the power-on or power-off command to the display using the CEC protocol. The first parameter specifies the output: 0 = HDBaseT; 1 = HDMI. The second parameter sets the power state of the display: 0 = off; 1 = on.

Syntax
CEC:Trig X Y

Parameter	Description	Range
X	Output	0, 1
Y	Mode	0, 1

Example

```
CEC:Trig 1 1
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "cec: trig 1 1"
}
```

Display:BYOD:Kick

Removes (“kicks”) the existing BYOD connection from the AT-OME-MS52W.

Syntax
Display:BYOD:Kick

This command does not require any parameters

Example

```
Display:BYOD:Kick
```

Feedback

```
{
  “result”: {
    “success”: true
  },
  “methodreturn”: “display: byod: kick”,
  “jsonrpc”: “2.0”
}
```

Display:Control:IP:Get

Returns the IP address and port of the remote IP device, such as a display.

Syntax
Display:Control:IP:Get

This command does not require any parameters

Example

```
Display:Control:IP:Get
```

Feedback

```
{
  “result”: {
    “port”: 23,
    “ip”: “10.20 .50 .58”
  },
  “methodreturn”: “display: control: ip: get”,
  “jsonrpc”: “2.0”
}
```

Display:Control:IP:Set

Sets the IP address and port of the remote IP device, such as a display. The IP address must be specified in dot-decimal notation.

Syntax
Display:Control:IP:Set X Y

Parameter	Description	Range
X	IP address of device	1 ... 254 (per octet)
Y	Port	0 ... 65535

Example

```
Display:Control:IP:Set 10.20.50.58 23
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "display: control: ip: set 10.20 .50 .58 23",
  "jsonrpc": "2.0"
}
```

Display:Input:All:Get

Returns a list of all inputs and the current status. If an input is not active, `false` is returned. Otherwise, `true` is returned. For BYOD devices, the protocol will be displayed in the `type` key. Inputs are represented numerically: 0 = USB-C, 1 = DisplayPort, 2 = HDMI, 3 = HDMI, 4 = BYOD.

Syntax
Display:Input:All:Get

This command does not require any parameters

Example

```
Display:Input:All:Get
```

Feedback

```
{
  "
  result": {
    "
    0": {
      "status": false
    },
    "1": {
      "status": true
    },
    "2": {
      "status": true
    },
    "3": {
      "status": false
    },
    "4": {
      "type": "airplay",
      "status": false
    }
  },
  "methodreturn": "display: input: all: get",
  "jsonrpc": "2.0"
}
```

Display:Input:Get

Returns the active input. If no input is connected, then `unknown` is returned.

Syntax
Display:Input:Get

This command does not require any parameters

Example

```
Display:Input:Get
```

Feedback

```
{
  "result": {
    "input": 4,
    "type": "airplay"
  },
  "methodreturn": "display: input: get",
  "jsonrpc": "2.0"
}
```

Display:Input:HDCP:State:Get

Returns the HDCP status of the specified input.

Syntax
Display:Input:HDCP:State:Get X

Parameter	Description	Range
X	Input	0 ... 4

Example

```
Display:Input:HDCP:State:Get 0
```

Feedback

```
{
  "result": {
    "state": true
  },
  "methodreturn": "display: input: hdcp: state: get 0"
}
```

Display:Input:HDCP:State:Set

Sets the HDCP state on the specified input. When specifying the second argument, 0 = off, 1= on.

Syntax
Display:Input:HDCP:State:Set X Y

Parameter	Description	Range
X	The specified input	0 ... 4
Y	The HDCP state	0, 1

Example

```
Display:Input:HDCP:State:Set 0 1
```

Feedback

```
{
  "result": {
    "success": true
  },
  "methodreturn": "display: input: hdcp: state: set 0 1"
}
```

Display:Input:Set

Sets the active input.

Syntax
Display:Input:Set input X

Parameter	Description	Range
X	The specified input	0 ... 4

Example

```
Display:Input:Set 1
```

Feedback

```
{
  "result": {
    "activeinput": 0
  },
  "methodreturn": "display: input: set 0",
  "jsonrpc": "2.0"
}
```

Display:InputState:Get

Returns the input state of the specified input. If 0 is returned, then no input is connected to the queried input. Otherwise, 1 will be returned for an input with a valid connection. Use values 0 through 4 to specifiy the input: 0 = USB-C, 1 = DisplayPort, 2 = HDMI, 3 = HDMI, 4 = BYOD.

Syntax
Display:InputState:Get X

Parameter	Description	Range
X	Input	0 ... 4

Example

```
Display:InputState:Get 1
```

Feedback

```
{
  "result": {
    "input": 1,
    "state": false
  },
  "methodreturn": "display: inputstate: get 1",
  "jsonrpc": "2.0"
}
```

Display:Matrix:Get

Returns the input for the specified output in Matrix Mode. Set the parameter to 0 to query the HDBaseT output or set to 1 to query the HDMI output.

Syntax
Display:Matrix:Get X

Parameter	Description	Range
X	The specified output	0, 1

Example

```
Display:Matrix:Get 1
```

Feedback

```
{
  "result": {
    "input": 4
  },
  "methodreturn": "display: matrix: get 1",
  "jsonrpc": "2.0"
}
```

Display:Matrix:Mode:Get

Returns the current Matrix Mode. If Matrix Mode is disabled, then the `mode` key will return `false`. When Matrix Mode is enabled, the `mode` key will return `true` and the mode will be displayed in the `subtype` key. Use the Display:Matrix:Mode:Set command to enable or disable Matrix Mode.

Syntax
Display:Matrix:Mode:Get

This command does not require any parameters

Example

```
Display:Matrix:Mode:Get
```

Returns

```
{
  "result": {
    "mode": true,
    "subtype": "MATRIX_MODE_STATIC"
  },
  "methodreturn": "display: matrix: mode: get",
  "jsonrpc": "2.0"
}
```

Display:Matrix:Mode:Set

Enables or disabled Matrix Mode. 0 = Disable Matrix Mode; 1 = Matrix Mode; 2 = Matrix Mode with Static Route.

Syntax
Display:Matrix:Mode:Set X

Parameter	Description	Range
X	State	0 ... 2

Example

```
Display:Matrix:Mode:Set 2
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "display: matrix: mode: set 2"
}
```

Display:Matrix:Set

Routes the specified input to the desired output. The first parameter is the input port: 0 = USB-C (port 1), 1 = DP IN (port 2), 2 = HDMI IN (port 3), 3 = HDMI IN (port 4), 4 = BYOD. The second parameter is the output port: 0 = HDBaseT, 1 = HDMI. If Matrix Mode is disabled, this command will return “Command Failure”.

Syntax
Display:Matrix:Set X Y

Parameter	Description	Range
X	Input port	0 ... 4
Y	Output port	0, 1

Example

```
Display:Matrix:Set 0 0
```

Returns

```
{
  “result”: {
    “success”: true
  },
  “methodreturn”: “Display: Matrix: Set 0 0”,
  “jsonrpc”: “2.0”
}
```

Display:Minimal:Get

Returns the state of the output display. If a display is not connected, then `false` will be returned. JSON structure is not used in the return value.

Syntax
Display:Minimal:Get

This command does not require any parameters

Example

```
Display:Minimal:Get
```

Returns

```
on
```

Display:Minimal:Set

Sets the state of the output display. If a display is not connected, then `false` will be returned.

Syntax
Display:Minimal:Set X

Parameter	Description	Range
X	State	0, 1

Example

```
Display:Minimal:Set 1
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "display: minimal: set 1"
}
```

GetBroadcast

Returns the current broadcast setting. JSON structure is not used in the return value.

Syntax
GetBroadcast

This command does not require any parameters

Example

```
GetBroadcast
Off
```

GetHostName

Returns the hostname of the AT-OME-MS52W.

Syntax
GetHostName

This command does not require any parameters

Example

```
GetBroadcast
```

Returns

```
{
  "result": {
    "hostname": "ms52w - 34e4"
  },
  "methodreturn": "gethostname"
}
```

GetInputStates

Returns the state of each input on the AT-OME-MS52W. JSON structure is not used in the return value. A value of `true` indicates that the input has a valid source connection. A value of `false` indicates no source connected.

Syntax
GetInputStates

This command does not require any parameters

Example

```
GetInputStates
```

Returns

```
[GetInputState 0 false, GetInputState 1 false, GetInputState 2 true, GetInputState 3 false, GetInputState 4 true]
```

GetNoSignalTimeout

Returns the current time interval which must elapse before the connected display is powered-off.

Syntax
GetNoSignalTimeout

Example

```
GetNoSignalTimeout
```

Returns

```
{
  "result": {
    "units": "seconds",
    "value": 0
  },
  "methodreturn": "getnosignaltimeout"
}
```

Help

Returns additional information about the specified command. JSON structure is not used in the return value.

Syntax
Help X

Parameter	Description	Range
X	Command name	String

Example

```
Help Display:Minimal:Set
```

Returns

```
Display:Minimal:Set <value>

    value - 0 = off

            1 = on
```

Input:Get

Returns the active input. If no input is connected, then `unknown` is returned.

Syntax
Input:Get

This command does not require any parameters

Example

```
Input:Get
```

Feedback

```
{
  "result": {
    "input": 4,
    "type": "airplay"
  },
  "methodreturn": "input: get",
  "jsonrpc": "2.0"
}
```

Input:Set

Sets the active input.

Syntax
Input:Set X

Parameter	Description	Range
X	The specified input	0 ... 4

Example

```
Input:Set 1
```

Feedback

```
{
  "result": {
    "activeinput": 0
  },
  "methodreturn": "input: set 0",
  "jsonrpc": "2.0"
}
```

Instruments:Temperature:Get

Returns the internal temperature of the AT-OME-MS52W. If the `units` parameter is not specified, then the temperature will be displayed in Celsius. Specify c for Celsius, f for Fahrenheit, or k for Kelvin.

Syntax
Instruments:Temperature:Get X

Parameter	Description	Range
X	Units (optional)	c, f, k

Example

```
Instruments:Temperature:Get
```

Returns

```
{
  "result": {
    "scale": "Celcius",
    "value": 50
  },
  "methodreturn": "instruments: temperature: get"
}
```

Misc:Model:Get

Returns the SKU of the AT-OME-MS52W.

Syntax
Misc:Model:Get

This command does not require any parameters

Example

```
Misc:Model:Get
```

Returns

```
{
  "result": {
    "model": "AT - OME - MS52W"
  },
  "methodreturn": "misc: model: get"
}
```

Misc:Version:Get

Returns the hardware version of the AT-OME-MS52W. If no parameters are specified, then the master firmware version is returned.

Syntax
Misc:Version:Get X Y

Parameter	Description	Range
X	Version	master, mcu

Example

```
Misc:Version:Get mcu
```

Returns

```
{
  "result": {
    "version": {
      "mcu": "1.0 .00"
    }
  },
  "methodreturn": "misc: version: get mcu"
}
```

Misc:Versions:Get

Returns both the MCU and master firmware versions of the AT-OME-MS52W.

Syntax
Misc:Versions:Get

This command does not require any parameters

Example

```
Misc:Versions:Get
```

Returns

```
{
  "result": {
    "versions": {
      "mcu": "V1 .0 .00",
      "master": "2.8 .1"
    }
  },
  "methodreturn": "misc: versions: get"
}
```

Moderator:Enable:Get

Returns the current status of Moderator mode. If Moderator Mode is disabled, then `false` is returned.

Syntax
Moderator:Enable:Get

This command does not require any parameters

Example

```
Moderator:Enable:Get
```

Returns

```
{
  "result": {
    "enable": true
  },
  "methodreturn": "moderator: enable: get",
  "jsonrpc": "2.0"
}
```

Moderator:Enable:Set

Enable or disable Moderator mode. Specify `1` to enable Moderator mode or `0` to disable. Two JSON structures are returned.

Syntax
Moderator:Enable:Set X

Parameter	Description	Range
X	State	0, 1

Example

```
Moderator:Enable:Set 1
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "moderator: enable: set 1"
}

{
  "
  jsonrpc": "2.0",
  "event": {
    "moderator": {
      "streams": [],
      "activeindex": -1,
      "enabled": true
    }
  }
}
```

Moderator:Kick

Kicks the specified BYOD ID from the system.

Syntax
Moderator:Kick X

Parameter	Description	Range
X	Client ID	Integer

Example

```
Moderator:Kick 2
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "Moderator: Kick 2"
}
```

Moderator:Show

Sets the active BYOD device ID for casting.

Syntax
Moderator:Show X

Parameter	Description	Range
X	Client ID	Integer

Example

```
Moderator:Show 2
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "Moderator: Show 2"
}
```

Moderator:Status:Get

Returns the current status of Moderator mode.

Syntax
Moderator:Status:Get

This command does not require any parameters

Example

```
Moderator:Status:Get
```

Returns

```
{
  "methodreturn": "moderator: status: get",
  "moderator": {
    "streams": [2],
    "activeindex": 1,
    "enabled": true
  },
  "jsonrpc": "2.0"
}
```

Net:GetInfo

Returns information about the physical and wireless network interfaces.

Syntax
Net:GetInfo

This command does not require any parameters

Example

```
Net:GetInfo
```

Returns


```
{
  "result": {
    "netinfo": {
      "wlan0": {
        "mode": "access point",
        "txbytes": 322684,
        "txpackets": 1119,
        "netmask": "255.255 .255 .0",
        "ip": "192.168 .43 .1",
        "rxbytes": 0,
        "mac": "74: da: 38: ef: db: e8",
        "rxpackets": 9841,
        "ap": {
          "channel": 165,
          "ssid": "ms52w - b500"
        }
      },
      "eth0": {
        "mode": "dynamic",
        "txbytes": 281994851,
        "txpackets": 1268641,
        "netmask": "255.255 .255 .0",
        "ip": "10.20 .20 .32",
        "rxbytes": 286555191,
        "mac": "00: 1 e: 06: 36: b5: 00",
        "rxpackets": 2845101
      }
    }
  },
  "methodreturn": "net: getinfo"
}
```

OutputMode

Sets the output mode for command feedback, where h = human-readable, j = JSON (compact), p = JSON (“pretty”). To display without using JSON data structures, specify the `h` parameter.

Syntax
OutputMode X

Parameter	Description	Range
X	Mode	h, j, p

Example

```
OutputMode h
```

Returns

```
OutputMode is Human Readable
```

The following examples illustrate the effect on command feedback, after executing the OSD:State:Get command.

h (human-readable)	j (JSON - compact)	p (JSON - “pretty”)
OSD:State:Get `OSD State is enabled`	OSD:State:Get `{ “result”: { “state”: true }, “methodreturn”: “osd: state: get” }`	OSD:State:Get `{“result”: {“state”: true}}`

Platform:Reset

Resets the AT-OME-MS52W to factory-default settings. Network settings are preserved, unless otherwise specified by the second parameter. This parameter is optional.

Syntax
Platform:Reset X Y

Parameter	Description	Range
X	System	all
Y	Network system (optional)	network

Example

```
Platform:Reset all
```

Returns

```
{
  "result": {
    "success": true
  },
  "jsonrpc": "2.0"
}
```

Platform:Restart

Reboots the AT-OME-MS52W.

Syntax
Platform:Restart

This command does not require any parameters

Example

```
Platform:Restart
```

Returns

```
{
  "result": {
    "success": true
  },
  "jsonrpc": "2.0"
}
```

Platform:Shutdown

Shuts down the power to the AT-OME-MS52W. This command should be performed before disconnecting the power from the unit.

Syntax
Platform:Shutdown

This command does not require any parameters

Example

```
Platform:Shutdown
```

Returns

```
{
  "result": {
    "success": true
  },
  "jsonrpc": "2.0"
}
```

Quit

Ends the Telnet session.

Syntax
Quit

This command does not require any parameters

Example

```
Quit
```

Returns

```
Connection closed by foreign host.
Session closed.
```

Relay:PulseDuration:Get

Returns the current pulse duration of the relay.

Syntax
Relay:PulseDuration:Get

This command does not require any parameters

Example

```
Relay:PulseDuration:Get
```

Returns

```
{
  "result": {
    "duration": 1
  },
  "methodreturn": "relay: pulseduration: get"
}
```

Relay:PulseDuration:Set

Sets the relay pulse duration.

Syntax
Relay:PulseDuration:Set

Parameter	Description	Range
Y	Duration (seconds)	1 ... 60

Example

```
Relay:PulseDuration:Set 2
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "relay: pulseduration: set 2"
}
```

Relay:State:Get

Returns the current state of the specified relay.

Syntax
Relay:State:Get X

Parameter	Description	Range
X	Relay number	0, 1

Example

```
Relay:State:Get 0
```

Returns

```
{
  "result": {
    "state": true
  },
  "methodreturn": "relay: state: get 0"
}
```

Relay:State:Set

Sets the state of the specified relay. The relay can be normally-open (NO) or normally-closed (NC).

Syntax
Relay:State:Set X Y

Parameter	Description	Range
X	Relay number	0, 1
Y	Relay state	0, 1

Example

```
Relay:State:Get 0 1
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "relay: state: set 0 1"
}
```

SetBroadcast

Enables or disables broadcast mode. When set to `on`, any state change to the AT-OME-MS52W will be reflected through RS-232, Telnet, and TCP port 9000. State changes can be caused by modification of the web server, physical connection/disconnection of source/sink and additional Telnet/TCP clients making changes. `1` = broadcast enabled; `0` = broadcast disabled. The default setting is `on`. JSON structure is not used in the return value.

Syntax
SetBroadcast X

Parameter	Description	Range
X	State	0, 1

Example

```
SetBroadcast 1
```

Returns

```
Broadcast on
```

SetVol

Shortened command for setting the audio volume commands. The command argument can be any one of the following:

`+` increment volume by 1 dB.
`-` decrement volume by 1 dB.
`sta` returns the current SetVol value.

Syntax
SetVol X

Parameter	Description	Range
X	Value	+, -, sta

Example

```
SetVol +
```

Returns

```
{
  "result": {
    "volume": 0,
    "success": true
  },
  "methodreturn": "setvol +"
}
```

USBRouting:Follow:Input

Assigns the video input with the specified USB host when using Follow USB mode.

Syntax
USBRouting:Follow:Input X Y

Parameter	Description	Range
X	Video input	1 ... 4
Y	USB host	1 ... 3 (Host 1, Host 2, Remote host)

Example

```
USBRouting:Follow:Input 2 1
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "usbrouting:follow:input 2 1"
}
```

USBRouting:Input:Get

Returns the USB host and corresponding input for the Manual USB Routing mode.

Syntax
USBRouting:Input:Get

This command does not require any parameters

Example

```
USBRouting:Input:Get
```

Returns

```
{
  "result": {
    "input": 1,
    "host": "1"
  },
  "methodreturn": "usbrouting:input:get",
  "jsonrpc": "2.0"
}
```

USBRouting:Input:Set

Sets the active USB host port.

1 USB host 1
2 USB host 2
3 Remote host
4 USB-C

Syntax
USBRouting:Input:Set X

Parameter	Description	Range
X	Value	1 ... 4

Example

```
USBRouting:Input:Set 2
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "usbrouting:input:set 2"
}
```

USBRouting:Mode:Get

Returns the USB routing mode.

Syntax
USBRouting:Mode:Get

This command does not require any parameters

Example

```
USBRouting:Mode:Get
```

Returns

```
{
  "result": {
    "mode": "AUTOSWITCH"
  },
  "methodreturn": "usbrouting:mode:get",
  "jsonr
  pc ":"
  2.0 "
}
```

USBRouting:Mode:Set

Sets the USB routing mode.

Syntax
USBRouting:Mode:Set X

Parameter	Description	Range
X	Value	MANUAL, AUTOSWITCH, FOLLOW

Example

```
USBRouting:Mode:Set manual
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "usbrouting:mode:set manual"
}
```

Zone:PortParams

Returns the current RS-232 settings for the specified zone. 1 = RS-232 port; 2 = HDBaseT OUT port.

Syntax
Zone:PortParams X

Parameter	Description	Range
X	The zone to query	1, 2

Example

```
Zone:PortParams 1
```

Returns

```
{
  "result": {
    "zone": 1,
    "params": "[9600, 8, 0, 1]"
  },
  "methodreturn": "zone: portparams 1"
}
```

Zone:PortSetup

Sets the RS-232 settings for the specified zone. 1 = RS-232 port; 2 = HDBaseT OUT port.

Syntax
Zone:PortSetup V W X Y Z

Parameter	Description	Range
V	The zone to assign settings to	1, 2
W	Baud rate	9600 ... 115200
X	Data bits	7, 8
Y	Parity bit	0, 1, 2
Z	Stop bit	1, 0

Example

```
Zone:PortSetup 1 19200 8 0 1
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "zone: portsetup 1 19200 8 0 1"
}
```

Zone:SendCmd

Sends an RS232 command to the specified zone. 1 = RS-232 port; 2 = HDBaseT OUT port.

Syntax
Zone:SendCmd X Y

Parameter	Description	Range
X	The zone to send to	1, 2
Y	Command	command

Example

```
Zone:SendCmd 1 vol+
```

Returns

```
{
  "result": {
    "success": true
  },
  "methodreturn": "zone: sendcmd 1 vol
    + "
}
```