

Muxlab Network Controller (MNC)

Software Version 1.5.6c

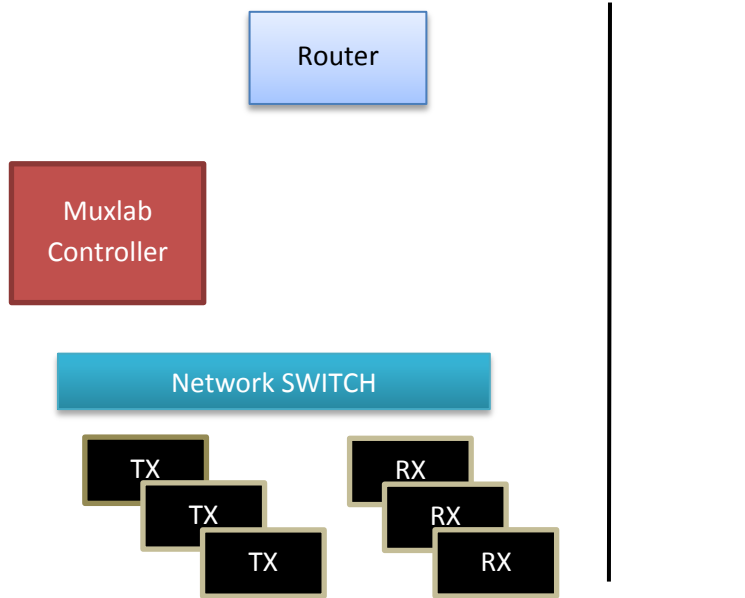
IP commands to control the following products via MNC:

- 500810/811 (MNC)
- 500752/500753/500754/500755/500756 (HDMI Over IP)
- 500480 (Matrix 16x16 4K60)
- 500758/500759 (HDMI Over IP 4K)
- 500757 (HDMI Over IP H264)
- 500762 (HDMI Over IP H.264 POE Extender)

Contents

Global Low level architecture.....	3
1. IP Command API: definition and format	4
1.1 Definition.....	4
1.2 General Format	4
2. IP Command API for 500752\753\754\755\756\762\758\759: command/response list.....	5
2.1 Get product list from the database.....	5
2.2 Automatic discovery.....	5
2.3 Manual discovery	5
2.4 Get devices from the database	6
2.5 Add Remote TX/Delete Remote TX (For 762)	6
2.6 Blink Link LED (For 762).....	7
2.7 Update some devices attributes	7
2.8 Reboot device.....	9
2.9 Connect/disconnect device	9
2.10 Select and apply a preset	10
2.11 Save current matrix connections in a specific preset.....	10
2.12 Save current matrix connections in a NEW preset name	10
2.13 Delete a preset	11
2.14 Send data to RS-232 (For 500753/754/755/756/757/758/759/762/758/759)	11
2.15 Send data to IR (For 500752/753/754/762 (TX) and 500755/756/762 (TX & RX)	11
2.16 Send HDMI CEC Command (FOR 758/759)	12
2.17 Modify network setting OF THE MNC	12
2.18 Modify Administrator password OF THE MNC.....	13
2.19 Get a list of users.....	13
2.20 Get a list of permissions for a user.....	13
2.21 VIDEO WALL: Connection command (For 500754/759)	14
2.22 VIDEO WALL: Select and apply a configuration (For 500754/500759)	15
2.23 VIDEO WALL: Changing source to a configuration (For 500754/500759).....	15
3. IP Command API for 500480: command/response list	16
3.1 Get all port status from the database	16
3.2 Get all presets	16
3.3 Connect/disconnect device.....	17
3.4 Update some ports attributes.....	17
3.5 Select and apply a preset	18
4. API Command Usage Guide	19
4.1 Software	19
4.2 Examples of Commands	20

Global Low level architecture



Network SWITCH

1. IP Command API: definition and format

1.1 Definition

The IP command API use HTTP POST with JSON data.

Each IP command must be sent to the following URL:

http://aaa.bbb.ccc.ddd/mnc/secure_api.php

Each IP command must contain the MNC username and password

aaa.bbb.ccc.ddd: the MNC IP address (default static IP: **192.168.168.50**)

p_userName: the MNC user name (default value: **admin**)

p_password: the MNC user password (default value: **admin**)

1.2 General Format

The IP command (JSON data) has the following format.

Command **without** data parameters:

```
{"p_targetId":<target id>,"p_cmd": "<command name>","p_userName": "<MNC User Name >","p_password": "<MNC password >"}
```

Response format **without** additional data:

```
{"p_targetId":<target id>,"p_cmd": "<command name>","p_rspStatus": "<command status>","p_msg": "<a message>"}
```

Command **with** data parameters:

```
{"p_targetId":<target id>,"p_cmd": "<command name>","p_userName": "<MNC User Name >","p_password": "<MNC password >","p_data": [{"key1": "value1", "key2": "value2", ... }]}
```

Response format **with** additional data:

```
{"p_targetId":<target id>,"p_cmd": "<command name>","p_rspStatus": "<command status>","p_msg": "<a message>","p_data": [{"key1": "value1", "key2": "value2", ... }]}
```

⇒ “p_data” field will depend on the associated command.

⇒ “p_targetId” value is:

“0” to send a command for the MNC

Use the system **ID** displayed in the “Product” web page.

This is not a constant value, it depends on the client system

EX: To set-up an HDMI Over IP Virtual matrix (must be done only once)

- Login to MNC
- On “products” web page, select “Hdmi Over IP (500752/753) and click “Add Device”
- Enter a name (ex: myHdmiOverIpMatrix) then click OK
- Your Hdmi Over IP virtual Matrix is now added in your system, its system **ID** is now displayed: You must use this ID in the IP commands.
- To auto-configure all 500752/753, click on “select “ then “Launch automatic discovery”

2. IP Command API for 500752\753\754\755\756\762\758\759: command/response list

2.1 Get product list from the database

Description:

The system will retrieve all the product lists currently saved in the database

Command:

```
{ "p_targetId": 0, "p_cmd": "getProductListSaved", "p_userName": "<MNC User Name>", "p_password": "<MNC password>" }
```

Response:

```
{ "p_targetId": 0, "p_cmd": "getProductListSaved", "p_rspStatus": "SUCCESS", "p_msg": "<a_message>", "p_data": [{"globalproductlistsavedID": "<value>", "ProductCustomName": "<value>", "ProductIp": "<IP Address>", "ProductMask": "<Netmask>", "ProductIsDhcp": "<0/1>", "ProductSupported_ProductSupportedID": "<value>", "ProductLogin": "<value>", "ProductPassword": "<value>", "ProductIdOverride": "<0/1>{ ... }, ...} }
```

2.2 Automatic discovery

Description:

The system will retrieve all the devices in the network and automatically send all the necessary updates to the devices to configure them properly (duplicate IP address will not be solved by the system; it will let the user resolve this kind of issue). Then it will return the list of devices found with their attributes.

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "launch_discovery_auto", "p_userName": "<MNC User Name>", "p_password": "<MNC password>" }
```

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "launch_discovery_auto", "p_rspStatus": "SUCCESS", "p_msg": "<a_message>", "p_data": [{"productName": "<value>", "modelName": "<value>", "customName": "<value>", "mac": "<value>", "ip": "<value>", "mask": "<value>", "isDhcp": "<0/1>", "multicastGroupIp": "<value>", "videoResolution": "<value>", "videoFrameRate": "<value>", "audioFormat": "<value>", "isVideoSignalDetected": "<0/1>", "isIrOn": "<0/1>", "isDipSwitchEnabled": "<0/1>", "fwVer": "<value>", "uartBaudRate": "<value>", "irMode": "<emitter/sensor>", "rs232FeedbackIP": "<value>", "irFeedbackIP": "<value>", "isRs232FeedbackOn": "<0/1>", "isRs232IpHeader": "<0/1>", "compressionRate": "<value>", "isAutoCompressionOn": "<0/1>", "is60fps": "<0/1>", "isDisplayConnected": "<0/1>", "isScreenImageOn": "<0/1>", "isScreenTextOn": "<0/1>", "connectedMac": "<value>", "isAutoResolutionOn": "<0/1> { ... }, ...} }
```

In **Red**, these are the additional attributes for RX device

In **Green**, these are the additional attributes for TX device

2.3 Manual discovery

Description:

The system will retrieve all the devices in the network and it will return the list of devices found with their attributes. (No other actions will be performed)

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "launch_discovery", "p_userName": "<MNC User Name>", "p_password": "<MNC password>" }
```

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "launch_discovery_auto", "p_rspStatus": "SUCCESS", "p_msg": "<a_message>", "p_data": [{"productName": "<value>", "modelName": "<value>", "customName": "<value>", "mac": "<value>", "ip": "<value>", "mask": "<value>", "isDhcp": <0/1>, "multicastGroupIp": "<value>", "videoResolution": "<value>", "videoFrameRate": "<value>", "audioFormat": "<value>", "isVideoSignalDetected": <0/1>, "isIrOn": <0/1>, "isDipSwitchEnabled": <0/1>, "fwVer": "<value>", "uartBaudRate": "<value>", "irMode": "<emitter/sensor>", "rs232FeedbackIP": "<value>", "irFeedbackIP": "<value>", "isRs232FeedbackOn": <0/1>, "isRs232IpHeader": <0/1>, "compressionRate": "<value>", "isAutoCompressionOn": <0/1>, "is60fps": <0/1>, "isDisplayConnected": <0/1>, "isScreenImageOn": <0/1>, "isScreenTextOn": <0/1>, "connectedMac": "<value>", "isAutoResolutionOn": <0/1> { ... }, ...] }
```

In **Red**, these are the additional attributes for RX device

In **Green**, these are the additional attributes for TX device

2.4 Get devices from the database

Description:

The system will retrieve all the devices currently stored in the database

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "get_devices", "p_userName": "<MNC User Name>", "p_password": "<MNC password>" }
```

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "launch_discovery_auto", "p_rspStatus": "SUCCESS", "p_msg": "<a_message>", "p_data": [{"productName": "<value>", "modelName": "<value>", "customName": "<value>", "mac": "<value>", "ip": "<value>", "mask": "<value>", "isDhcp": <0/1>, "multicastGroupIp": "<value>", "videoResolution": "<value>", "videoFrameRate": "<value>", "audioFormat": "<value>", "isVideoSignalDetected": <0/1>, "isIrOn": <0/1>, "isDipSwitchEnabled": <0/1>, "fwVer": "<value>", "uartBaudRate": "<value>", "irMode": "<emitter/sensor>", "rs232FeedbackIP": "<value>", "irFeedbackIP": "<value>", "isRs232FeedbackOn": <0/1>, "isRs232IpHeader": <0/1>, "compressionRate": "<value>", "isAutoCompressionOn": <0/1>, "is60fps": <0/1>, "isDisplayConnected": <0/1>, "isScreenImageOn": <0/1>, "isScreenTextOn": <0/1>, "connectedMac": "<value>", "isAutoResolutionOn": <0/1> { ... }, ...] }
```

2.5 Add Remote TX/Delete Remote TX (For 762)

Description:

The system will add/delete a remote source

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "add_remote_tx", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "data": [{"customNameAdd": "<value>", "modelAdd": "<value>", "macAddressAdd": "RX-500762", "ipAddressAdd": "<value>", "ch1_RtspUrllanAdd": "<value>", "ch1_HlsUrllanAdd": "<value>", "ch1_TsUrllanAdd": "<value>", "ch1_FlvUrllanAdd": "<value>", "ch1_multicastIpAddress": "<value>", "ch1_multicastPortAdd": "<value>", "ch1_isHlsOnAdd": <0/1>, "ch1_isRtspOnAdd": <0/1>, "ch1_isTsOnAdd": <0/1>, "ch1_isFlvOnAdd": "<0/1>", "ch1_isMulticastOnAdd": "<0/1>"}] }
```

```
{ "p_targetId": <systemID>, "p_cmd": "delete_remote_tx", "p_userName": "<MNC User Name>", "p_password": "<MNC password >", "data": [{"mac": "<value>"}] }
```

Response:

```
{ "p_targetId":<systemID>,"cmd":"add_remote_tx","login_ack":"OK","pwd_msg":"SUCCESS",  
,"data": [{"customNameAdd": "<value>","modelAdd": "<value>","macAddressAdd": "<value>","  
ipAddressAdd": "<value>","ch1_RtspUrllanAdd": "<value>","ch1_HlsUrllanAdd": "<value>","  
"ch1_TsUrllanAdd": "<value>","ch1_FlvUrllanAdd": "<value>","ch1_multicastIpAdd": "<val  
ue>","ch1_multicastPortAdd": <value>,"ch1_isHlsOnAdd": <0/1>,"ch1_isRtspOnAdd": <0/1>,  
"ch1_isTsOnAdd": <0/1>,"ch1_isFlvOnAdd": "<0/1>","ch1_isMulticastOnAdd": "<0/1>"}] }  
{ "targ": "<value>","cmd":"delete_remote_tx","login_ack":"OK","pwd_msg":"SUCCESS","da  
ta": [{"mac": "<value>"}] }
```

2.6 Blink Link LED (For 762)

Description:

The system will send a blink command to the device. The device will blink the Link LED indefinitely unless it receives a value=0 blink command or if the device reboots.

Command:

```
{ "p_targetId":<systemID>,"p_cmd":"link_led","p_userName": "<MNC User Name  
>","p_password": "<MNC password >","data": [{"mac": "<value>","value": "<0/1>"}] }
```

Response:

```
{ "p_targetId":<systemID>,"cmd":"link_led","login_ack":"OK","pwd_msg": "SUCCESS", "dat  
a": [{"value": "<0/1>"}] }
```

2.7 Update some devices attributes

Description:

The system will update the devices specified with the new attributes provided. **Note that to update a device parameter, the device MUST already exist in the MNC database.**

Command:

```
{ "p_targetId":<systemID>,"p_cmd": "update_devices", "p_userName": "<MNC User  
Name>","p_password": "<MNC password>","p_data": [{"mac": "<device mac address>","attribute  
name":<attribute value>,..."}, {"mac": "<device mac address>","attribute name":<attribute value>,..."}] }
```

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >"
- o "portOut": "<output port number >"

List of attribute names that can be modified:

For TX and RX:

```
"customName" : set the custom name to give to this device  
"ip" : set the device ip address (EX: "192.168.1.80")  
"mask" : set the device mask (EX: "255.255.255.0")  
"isDhcp" : set the dhcp on(1) or off(0) (EX: 1 )  
"isIrOn" : set the IR on(1) or off(0) (EX: 1 )  
"isDipSwitchEnabled" : set the dip switch on(1) or off(0) (EX: 0 )  
"uartBaudRate" : set the RS232 baud rate (9600,19200,38400,57600,115200)  
"rs232FeedbackIP" : set IP address where RS232 data feedback should be sent  
"isRs232FeedbackOn" : To enable or disable data feedback  
"irMode" (500756 ONLY): set IR direction mode <sensor> OR <emitter>
```

For RX only:

"isScreenImageOn" : 1 => to display loaded image when no video signal
0 => to display a plain black screen when no video signal

"isScreenTextOn" : 1 => to display some debug text information
0 => to not display any debug text information

"videoResolution" : set the desired output video resolution
value="0051" to set video resolution to: 720X480 60Hz
value="2011" to set video resolution to: 480P 60Hz
value="2021" to set video resolution to:576P 50Hz
value="2034" to set video resolution to:720P 50Hz
value="2035" to set video resolution to:720P 60Hz
value="2041" to set video resolution to:1080P 24Hz
value="2042" to set video resolution to:1080P 25Hz
value="2044" to set video resolution to:1080P 50Hz
value="2045" to set video resolution to:1080P 60Hz
value="3041" to set video resolution to: 1080I 50Hz
value="3042" to set video resolution to: 1080I 60Hz

"isAutoResolutionOn" : 1 => to let the device auto-set video resolution
0 => to allow setting video resolution manually

"irFeedbackIP" : set IP address where IR data feedback should be sent

For TX only:

"isAutoCompressionOn" : 1 => to enable automatic video compression
0 => to disable automatic video compression

"irFeedbackIP" (500756 ONLY) : set IP address where IR data feedback should be sent

"is60fps" : 1 => to set video mode to 60 frames per second
0 => to set video mode to 30 frames per second

For 755-AMP only:

"volumeLevel" : 0 to 87 => Volume

"bassLevel" : -14 to 14 => Bass level

"trebleLevel" : -14 to 14 => Treble

"inputSelect" : 0 => Input #1
1 => Input #2
2 => Micro only
3 => Input #1 + Micro
4 => Input #2 + Micro

"microGain" : 0 to 26 => Microphone gain

"microBass" : -14 to 14 => Microphone bass

"microTreble" : -14 to 14 => Microphone treble

"secVolAtt" : -20 to 0 => Secondary volume attenuation

"micType" : 0 => Microphone electret
1 => Microphone dynamic
2 => Microphone ceramic

"micDetectionLevel" : 0 to 63 => Microphone detection level

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "update_devices", "p_rspStatus": "SUCCESS", "p_msg": "<am
message>", "p_data": [{"productName": "<value>", "modelName": "<value>", "customName": "<value>", "mac": "<value>"
, "ip": "<value>", "mask": "<value>", "isDhcp": <0/1>, "multicastGroupIp": "<value>", "videoResolution": "<value>", "videoFrameR
ate": "<value>", "audioFormat": "<value>", "isVideoSignalDetected": <0/1>, "isIrOn": <0/1>, "isDipSwitchEnabled": <0/1>, "isDisp
layConnected": <0/1>, "isScreenImageOn": <0/1>, "isScreenTextOn": <0/1>, "connectedMac": "<value>"}, { ... }, ... ] }
```


		500752	500753/754	500755	500756	500758/759	500762
RX & TX	customName	X	X	X	X	X	X
	ip	X	X	X	X	X	X
	mask	X	X	X	X	X	X
	isDhcp	X	X	X	X	X	X
	isIrOn	X	X	X	X		X
	isDipSwitchEnabled	X	X	X	X	X	X
	uartBaudRate	n/a	X	X	X	X	X
	rs232FeedbackIP	n/a	X	X	X	n/a	X
	isRs232FeedbackOn	n/a	X	X	X	n/a	X
	irMode	n/a	n/a	X	X	n/a	X
RX	isScreenImageOn	X	X	n/a	X	n/a	
	isScreenTextOn	X	X	n/a	X	n/a	
	videoResolution	X	X	n/a	X	X	
	isAutoResolutionOn	X	X	n/a	X	X	X
	irFeedbackIP	X	X	X	X	n/a	X
	hdcpFormat	n/a	n/a	n/a	n/a	X	
TX	isAutoCompressionOn	X	X	X	X	n/a	
	irFeedbackIP	n/a	n/a	n/a	X	n/a	X
	is60fps	X	X	n/a	X	n/a	

2.8 Reboot device

Description:

The system will reboot the devices

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "reboot_devices", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"mac": "<device mac address>"}, {...}, ...]}
```

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >"
- o "portOut": "<output port number >"

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "update_devices", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"mac": "<device mac address>", "rspStatus": "SUCCESS or FAILED", "msg": ""}, ...]}
```

2.9 Connect/disconnect device

Description:

Perform a connect/disconnect between devices

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "connection", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"macRx": "<Rx device mac address>", "macTx": "<Tx device mac address>"}, {...}, ...]}
```

"macRx": the RX mac address to connect/disconnect

"macTx": - To disconnect, use "00-00-00-00-00-00"

- To connect, use the TX device mac address to connect to

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >" (use "0" to disconnect the output port)
- o "portOut": "<output port number >"

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "connection", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"macRx": "<Rx device mac address>", "macTx": "<Tx device mac address>", "rspStatus": "SUCCESS or FAILED", "msg": ""}, ...]}
```

2.10 Select and apply a preset

Description:

Apply a (existent) preset

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "select_preset", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"presetId": "<preset id number>"}]}
```

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "select_preset", "p_rspStatus": "SUCCESS", "p_msg": "<message>", "p_data": [{"macRx": "<Rx device mac address>", "macTx": "<Tx device mac address>", "rspStatus": "SUCCESS or FAILED", "msg": ""}, ...]}
```

Note:

"p_data" will return all the connections result that took place

2.11 Save current matrix connections in a specific preset

Description:

Save the current matrix connections in a specific (existent) preset

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "save_preset", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"presetId": "<preset id number>"}]}
```

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "save_preset", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"presetId": "<preset ID number>"}]}
```

2.12 Save current matrix connections in a NEW preset name

Description:

Save the current matrix connections in a NEW preset name

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "create_preset", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"presetName": "<a new preset name>"}]}
```

Response:

```
{"p_targetId": <systemID>, "p_cmd": "create_preset", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"prestName": "<name of the preset>", "presetId": "<preset ID number>"}]}
```

2.13 Delete a preset

Description:

Delete a preset

Command:

```
{"p_targetId": <systemID>, "p_cmd": "delete_preset", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"presetId": "<preset id number>"}]}
```

Response:

```
{"p_targetId": <systemID>, "p_cmd": "delete_preset", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"presetId": "<preset ID number>"}]}
```

2.14 Send data to RS-232 (For 500753/754/755/756/757/758/759/762/758/759)

Description:

Send data (in hexadecimal) to RS232 port of a device

Command:

```
{"p_targetId": <systemID>, "p_cmd": "send_data_to_rs232", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"mac": "<Tx/Rx device mac address>", "hexdata": "<hex data>", "ack": "<0/1>", "feedback": "<0/1>"}]}
```

<hex data> maximum length = 390 characters (i.e: 195 bytes)

<feedback> 1 : to wait for a feedback data to be returned from the RS-232 port

<ack> 1 : to wait to wait for a command acknowledge

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >"
- o "portOut": "<output port number >"

Response:

```
{"p_targetId": <systemID>, "p_cmd": "send_data_to_rs232", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"mac": "<device mac address>", "rspStatus": "SUCCESS or FAILED", "msg": "", ...}]}
```

2.15 Send data to IR (For 500752/753/754/762 (TX) and 500755/756/762 (TX & RX))

Description:

Send data (in hexadecimal) to IR port of a device

Command:

```
{"p_targetId": <systemID>, "p_cmd": "send_data_to_ir", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"mac": "<Tx/Rx device mac address>", "hexdata": "<hex data>"}]}
```

<hex data> maximum length = 390 characters (i.e: 195 bytes)

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >"
- o "portOut": "<output port number >"

Response:

```
{ "p_targetId": <systemID>, "p_cmd": "send_data_to_ir", "p_rspStatus": "SUCCESS", "p_msg": "<a message>", "p_data": [{"mac": "<device mac address>", "rspStatus": "SUCCESS or FAILED", "msg": ""}, ...]}
```

2.16 Send HDMI CEC Command (FOR 758/759)

Description:

Send CEC command to a device

Command:

```
{ "p_targetId": <systemID>, "p_cmd": "send_hdmi_cec_command", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "data": [{"mac": "<value>", "command": "<value>"}]}
```

Command Values:

"standBy", "powerOn", "volumeUp", "volumeDown", "volumeMute"

Response:

```
{ "p_targetId": "<systemID>", "cmd": "send_hdmi_cec_command", "login_ack": "OK", "pwd_msg": "SUCCESS", "data": [{"mac": "<value>", "command": "<value>"}]}
```

2.17 Modify network setting OF THE MNC

Description:

Modify any network settings of the MNC (ip/mask/gateway/dhcp)

Command:

```
{ "p_targetId": 0, "p_cmd": "modifyNetSettings", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"dhcp": "<0/1>", {"ip": "<ip address>", {"mask": "<mask address>", {"gateway": "<gateway address>"}}]}
```

Response:

```
{ "p_targetId": 0, "p_cmd": "modifyNetSettings", "p_rspStatus": "SUCCESS/FAILED", "p_msg": "<a message>"}
```

2.18 Modify Administrator password OF THE MNC

Description:

Modify the administrator password of the MNC

Command:

```
{"p_targetId":0,"p_cmd":"changeAdminPswd","p_userName":"<MNC User Name>","p_password":"<MNC password>","p_data":[{"username":"<the user name>"}, {"currpswd":"<current passsword>"}, {"newpswd":"<new password>"}]}
```

Response:

```
{"p_targetId":0,"p_cmd":"modifyNetSettings","p_rspStatus":"SUCCESS/FAILED","p_msg":"<a message>"}
```

2.19 Get a list of users

Description:

Show a list of users on the MNC

Command:

```
{"p_targetId":0,"p_cmd":"getUsers","p_userName":"<MNC User Name>","p_password":"<MNC password>"}
```

Response:

```
{"p_targetId":0,"p_cmd":" getUsers","p_rspStatus":"SUCCESS","p_msg":"<a message>","p_data":[{"userID":"1","UserName":"<value>","UserEmail":"<value>","UserCreationDate":"<date time>","UserExpirationDate":"<datetime>","UserIsActivated":"0/1","UserRight_UserRightID":"0/1"},...]}
```

2.20 Get a list of permissions for a user

Description:

Show the permissions for a user on the MNC

Command:

```
{"p_targetId":0,"p_cmd":" getUserDetail","p_userName":"<MNC User Name>","p_password":"<MNC password>","p_data":[{"userid":"<the user id>"}]}
```

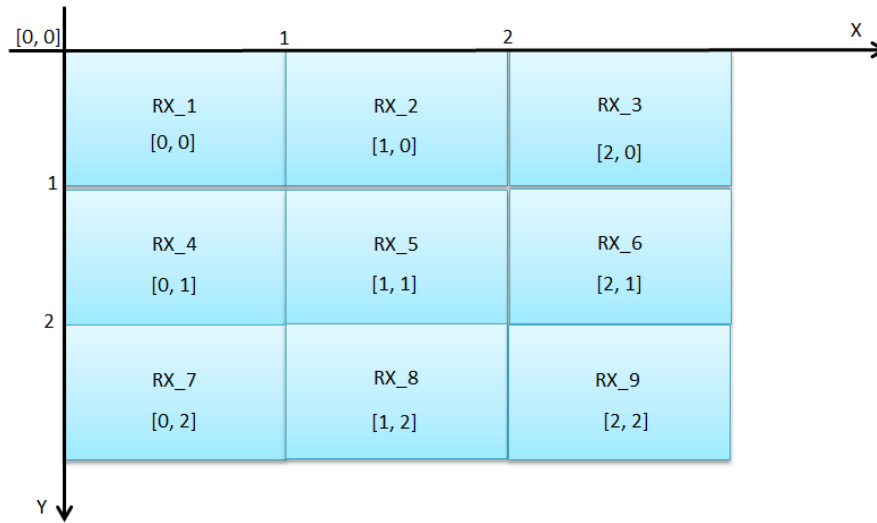
Response:

```
{"p_targetId":0,"p_cmd":" getUserDetail ","p_rspStatus":"SUCCESS","p_msg":"<a message>","p_data":[{"userID":"1","UserName":"<value>","UserIsActivated":"0/1","user rightID":"0/1"},...]}
```

2.21 VIDEO WALL: Connection command (For 500754/759)

Description:

Set RX as video wall and connect it to a source (TX)



Command:

```
{ "p_targetId": <systemID>, "p_cmd": "vwconnection", "p_userName": "<MNC User Name>", "p_password": "<MNC password>", "p_data": [{"macRx": "<Rx device mac address>", "macTx": "<Tx device mac address to connect to>", "isVideoWallMode": true, "vwGlobalSize": "<Global nb of TV on x axis>x<global nb of TV on y axis>", "vwSizeX": "<nb of TV of current video wall on x axis>", "vwSizeY": "<nb of TV of current video wall on y axis>", "xCoordinate": "<x coordinate of this RX>", "yCoordinate": "<y coordinate of this RX>", "measureUnit": "<inch or cm>", "screenWidth": "<screen width>", "screenHeight": "<screen Height>", "bezelTop": "<measure of top Tv bezel>", "bezelBottom": "<measure of bottom Tv bezel>", "bezelLeft": "<measure of left Tv bezel>", "bezelRight": "<measure of right Tv bezel>", "scaleUpH": "<0 to 255>", "scaleDownH": "<0 to 255>", "scaleUpV": "<0 to 255>", "scaleDownV": "<0 to 255>", "shiftRight": "<0 to 255>", "shiftLeft": "<0 to 255>", "shiftUp": "<0 to 255>", "shiftDown": "<0 to 255>", ...}] }
```

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn": "<input port number >"
- o "portOut": "<output port number >"

EX: Create a 2x1 video wall inside a global video wall of 2x2

```
{ "p_targetId": 1, "p_cmd": "vwconnection", "p_userName": "admin", "p_password": "admin", "p_data": [{"macRx": "00-0B-78-00-74-14", "macTx": "00-0B-78-00-74-15", "isVideoWallMode": true, "vwGlobalSize": "2x2", "vwSizeX": 2, "vwSizeY": 1, "xCoordinate": 0, "yCoordinate": 0, "measureUnit": "inch", "screenWidth": 34.8, "screenHeight": 19.6, "bezelTop": 0.1, "bezelBottom": 0.1, "bezelLeft": 0.1, "bezelRight": 0.1, "scaleUpH": 0, "scaleDownH": 0, "scaleUpV": 0, "scaleDownV": 0, "shiftRight": 0, "shiftLeft": 0, "shiftUp": 0, "shiftDown": 0}, ...] }
```

Response:

```
{"p_targetId":<systemID>,"p_cmd":"vwconnection","p_rspStatus":"SUCCESS","p_msg":"<a message>","p_data":[{"macRx":"<Rx device mac address>","macTx":"<Tx device mac address>","rspStatus":"SUCCESS or FAILED","msg":""}, ...]}
```

2.22 VIDEO WALL: Select and apply a configuration (For 500754/500759)

Description:

Apply a (existent) video wall configuration

Command:

```
{"p_targetId":<systemID>,"p_cmd":"vwApplyConfig ","p_userName": "<MNC User Name>","p_password": "<MNC password>","p_data":[{"vwConfigId ":"<vwConfig id number>"}]}
```

Response:

```
{"p_targetId":<systemID>,"p_cmd":"vwApplyConfig","p_rspStatus":"SUCCESS","p_msg":"<a message>","p_data":[{"macRx":"<Rx device mac address>","macTx":"<Tx device mac address>","rspStatus":"SUCCESS or FAILED","msg":""}, ...]}
```

Note:

"p_data" will return all the connections result that took place

2.23 VIDEO WALL: Changing source to a configuration (For 500754/500759)

Description:

Apply another source (TX) to an existent video wall configuration (But the original configuration parameters is not changed)

Command:

```
{"p_targetId":<systemID>,"p_cmd":"vwApplyConfig ","p_userName": "<MNC User Name>","p_password": "<MNC password>","p_data":[{"vwConfigId ":"<vwConfig id number>","macTx": "Tx device mac address"}]}
```

Note:

You have the option to use a port ID (to be defined in web server) instead of the mac address

- o "portIn":"<input port number >"

Response:

```
{"p_targetId":<systemID>,"p_cmd":"vwApplyConfig","p_rspStatus":"SUCCESS","p_msg":"<a message>","p_data":[{"macRx":"<Rx device mac address>","macTx":"<Tx device mac address>","rspStatus":"SUCCESS or FAILED","msg":""}, ...]}
```

Note:

"p_data" will return all the connections result that took place

3. IP Command API for 500480: command/response list

3.1 Get all port status from the database

Description:

The system will retrieve all the current port status

Command:

```
{"p_targetId":<systemID>,"p_cmd":"get_devices","p_userName":"<MNC User Name>","p_password":"<MNC password>"}
```

Response:

```
{"p_targetId":<systemID>,"p_cmd":"get_devices","p_rspStatus":"SUCCESS","p_msg":"<a_message>","p_data":[{"customName":"<value>","portId":"<value>","connectedPortId":"<value>","typeId":<value>,"typeName":"<value>","purposeId":"<value>","isPresent":<0/1>,"cardNb":"<value>"},{ ... }, ...]}
```

Note:

connectedPortId: "0" => disconnected

typeId:

1 => HDMI

2 => HDBT

3 => RS232

4 => VGA

5 => LAN

6 => IR

7 => USB

purposeId:

1 => INPUT_VIDEO

5 => OUTPUT_VIDEO

3.2 Get all presets

Description:

The system will retrieve all the preset ID and name

Command:

```
{"p_targetId":<systemID>,"p_cmd":"get_presets","p_userName":"<MNC User Name>","p_password":"<MNC password>"}
```

Response:

```
{"p_targetId":<systemID>,"p_cmd":"get_presets","p_rspStatus":"SUCCESS","p_msg":"<a_message>","p_data":[{"presetId":"<value>","presetIdv":"<value>","presetName":"<value>"},{ ... }, ...]}
```


3.3 Connect/disconnect device

Description:

Perform a connect/disconnect

Command:

```
{"p_targetId":<systemID>,"p_cmd":"connection","p_userName": "<MNC User Name>","p_password": "<MNC password>","p_data":[{"portOut": "<output #>","portIn": "<input #>"}, {...}, ...]}
```

- To disconnect, set input # = "0"

Response:

```
{"p_targetId":<systemID>,"p_cmd":"connection","p_rspStatus": "SUCCESS","p_msg": "<a message>","p_data": [{"portOut": "<output #>","portIn": "<input #>","rspStatus": "SUCCESS or FAILED","msg": ""}, ...]}
```

3.4 Update some ports attributes

Description:

The system will update the ports specified with the new attributes provided.

Command:

```
{"p_targetId":<systemID>,"p_cmd":"update_devices","p_userName": "<MNC User Name>","p_password": "<MNC password>","p_data":[{"port to update": "<port #>","attribute name":<attribute value>,..."}, {"port to update": "<port #>","attribute name":<attribute value>,...},...]}
```

Values of **<port to update>**: "portIn" or "portOut "

List of attribute names that can be modified:

For TX and RX:

"customName" : set the custom name to give to this port

Response:

```
{"p_targetId":<systemID>,"p_cmd":"update_devices","p_rspStatus": "SUCCESS","p_msg": "<amessage>","p_data": [{"customName": "<value>","portId": "<value>","connectedPortId": "<value>","typeId": "<value>","typeName": "<value>","purposeId": "<value>","isPresent": <0/1>,"card Nb": "<value>"}, { ... }, ...]}
```

3.5 Select and apply a preset

Description:

Apply a preset

Command:

```
{"p_targetId":<systemID>,"p_cmd":"select_preset","p_userName":"<MNC User Name>","p_password":"<MNC password>","p_data":[{"presetId":"<preset id number>"}]}
```

Response:

```
{"p_targetId":<systemID>,"p_cmd":"select_preset","p_rspStatus":"SUCCESS",  
"p_msg":"<amessage>","p_data":[{"portOut":"5-<output #>","portIn":"1-<input #>","rspStatus  
":"SUCCESS or FAILED","msg":""}, ...]}
```

Note:

"p_data" will return all the connections result that took place

4. API Command Usage Guide

The 500811 IP command API uses HTTP POST with JSON data. Each IP command must contain the MNC username and password. Two popular programs to send commands are Postman and cURL.

4.1 Software

Postman

The Postman Rest Client is a very popular and easy to use HTTP Request composer that makes it easy to call web services.

Install/Download:

Mac/Windows/Linux app: <https://www.getpostman.com/>

Instructions:

In a new tab:

- Change the dropdown to "POST" (default is "GET")
- Modify URL to "http://192.168.168.50/mnc/secure_api.php".
- Note: 192.168.168.50 is the default IP address of the 500811.**
- Under the Body tab, change to "raw" and "JSON (application/json)"
- In the text area, enter the command and click "Send"
- If successful, a response to the command will be sent back.

Command Examples:

This will return a list of devices from the 1st product target. p_targetId is ID number from the Product page:

```
{"p_targetId":1,"p_cmd":"get_devices","p_userName":"admin","p_password":"admin"}
```

This will send a hex data to an IR port of a device (port #1):

```
{"p_targetId":1,"p_cmd":"send_data_to_ir","p_userName":"admin","p_password":"admin","p_data":{"port":"1","hexdata":"0000006f"}}
```

This will apply another source (port #3) to an existent video wall configuration (vwConfigID #2):

```
{"p_targetId":1,"p_cmd":"vwApplyConfig","p_userName":"admin","p_password":"admin","p_data":{"vwConfigId":"2","portIn":"3"}}
```

cUrl

cURL is a computer software project providing a library and command-line tool for transferring data using various protocols.

Install/Download:

Linux/Mac: command-line tool

cURL Examples:

Note: 192.168.168.50 is the default IP address of the 500811. P_targetId is ID number from the Product page.

This will return a list of devices from the 1st product target:

```
curl -i -X POST -H 'Content-Type: application/json' -d  
'{"p_userName":"admin","p_password":"admin","p_targetId":1,"p_cmd":"get_devices"}'  
http://192.168.168.50/mnc/secure\_api.php
```

This will apply Preset #2:

```
curl -i -X POST -H 'Content-Type: application/json' -d  
'{"p_userName":"admin","p_password":"admin","p_cmd":"select_preset","p_targetId":1,"p_data":{"presetId":"2"}}'  
http://192.168.168.150/mnc/secure\_api.php
```

This will apply another source (port #5) to an existent video wall configuration (vwConfigID #1):

```
curl -i -X POST -H 'Content-Type: application/json' -d  
'{"p_userName":"admin","p_password":"admin","p_cmd":"vwApplyConfig","p_targetId":1,"p_data":{"vwConfigId":"1",  
"portIn":"5"}}' http://192.168.168.50/mnc/secure\_api.php
```

4.2 Examples of Commands

Muxlab Network Controller(MNC):

IP address: 192.168.168.50

Login: admin

Password: admin

Target ID: 1 (Click on "Product" menu in the 500811 web server to know the target ID to use)

Transmitter #1 Mac address: 00-0B-78-00-70-19 (assigned to port# 1 via 500811 web server)

Receiver #1 Mac address: 00-0B-78-00-75-38 (assigned to port# 1 via 500811 web server)

Receiver #2 Mac address: 00-0B-78-00-74-14 (assigned to port# 2 via 500811 web server)

All commands must be HTTP POST and sent to the following URL:

The data must be formatted in JSON, here are some command data examples:

- a) {"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data":{"macRx":"00-0B-78-00-75-38","macTx":"00-0B-78-00-70-19"}}
- b) {"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data":{"portOut":"2","portIn":"1"}}
- c) {"p_targetId":1,"p_cmd":"send_data_to_rs232","p_userName":"admin","p_password":"admin","p_data":{"portOut":"2","hexdata":"5F45Ac9B","ack":1,"feedback":1}}
- d) {"p_targetId":1,"p_cmd":"send_data_to_ir","p_userName":"admin","p_password":"admin","p_data":{"portIn":"1","hexdata":"AABBCC01020355FF"}}
- e) {"p_targetId":1,"p_cmd":"select_preset","p_userName":"admin","p_password":"admin","p_data":{"presetId":"6"}}
- f) {"p_targetId":1,"p_cmd":"update_devices","p_userName":"admin","p_password":"admin","p_data":{"irFeedbackIP":"192.168.168.50","portOut":"1"}}
- g) {"p_targetId":1,"p_cmd":"vwApplyConfig","p_userName":"admin","p_password":"admin","p_data":{"vwConfigId":"3"}}
- h) {"p_targetId":1,"p_cmd":"vwApplyConfig","p_userName":"admin","p_password":"admin","p_data":{"vwConfigId":"3","portIn":"1"}}