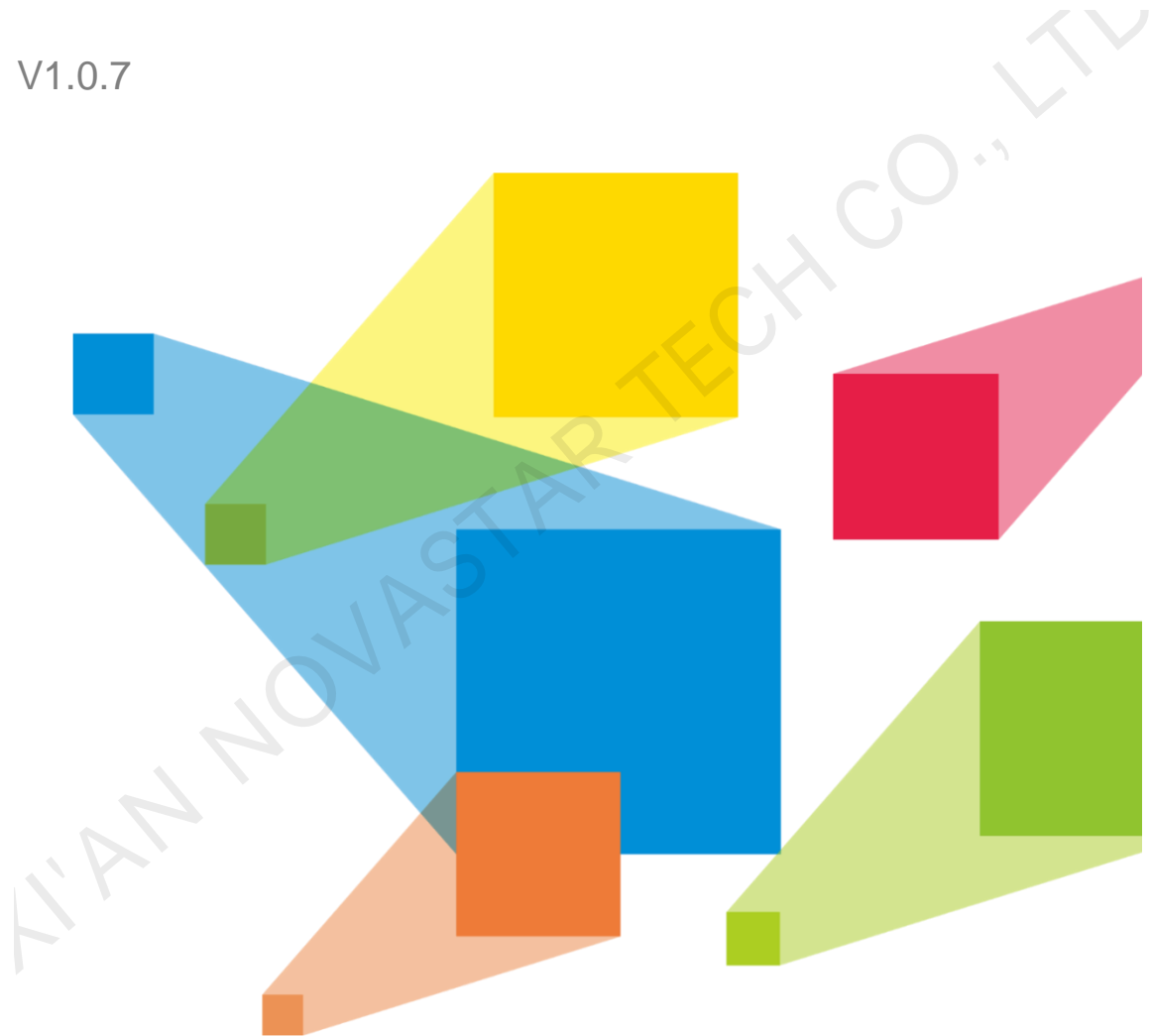


# H Series

## Video Wall Splicers

V1.0.7



Control Protocol

## Change History

Version	Modified By	Description	Modified On	Remarks
Ver1.0.0	Zhang Conghui	Initialized the control protocol.	2019-10-22	
Ver1.0.1	Zhang Conghui	Changed the "cmd" format.	2019-11-21	
Ver1.0.2	Zhang Zenglin	Added the read instructions.	2020-10-8	
Ver1.0.3	Zhang Zenglin	Added the instructions for adjusting screen color and MVR.	2020-11-8	
Ver1.0.4	Zhang Zenglin	Added the instructions for resetting to factory settings.	2020-11-15	
Ver1.0.5	Pan Xiufang	Changed the instructions for configuring IPC sources.	2021-03-10	
Ver1.0.6	Pan Xiufang	Added the instructions in Appendix B for configuring IPC mosaic sources.	2021-03-11	
Ver1.0.7	Zhang Zenglin	Added the instructions for saving screen brightness.	2021-03-18	

# Contents

Contents .....	ii
1. Communication Method .....	1
1.1 UDP Communication .....	1
1.2 Serial Communication .....	1
2. Command Format .....	1
3. Parameter Descriptions .....	4
4. Command Protocols .....	4
4.1 Layer Operations .....	4
4.1.1 Open Layers .....	4
4.1.2 Close Layers .....	5
4.1.3 Adjust Layer Position and Size .....	6
4.1.4 Bring to Front / Send to Back .....	6
4.1.5 Flip Layers .....	6
4.1.6 Switch Layer Input Sources .....	7
4.1.7 Crop Layer Input Sources .....	7
4.2 Preset Operations .....	7
4.2.1 Load Presets .....	7
4.2.2 Get Preset Enum .....	8
4.2.3 Clear Presets .....	8
4.2.4 Save Presets .....	8
4.3 Device Operations .....	9
4.3.1 Get Device Details .....	9
4.3.2 Get Slot Information .....	10
4.3.3 Get Connector Information .....	13
4.3.4 Reset to Factory Settings .....	14
4.4 Screen Operations .....	14
4.4.1 Adjust Screen Brightness (for screens loaded by LED 4K sending cards) .....	15
4.4.2 Set Screen FTB .....	15
4.4.3 Get Screen Output Information .....	15
4.4.4 Screen Layer Enum .....	16
4.4.5 Screen Color .....	17
4.4.6 Screen 3D (for screens loaded by LED 4K sending cards and for V1.3.0.0 or later version of the H series video wall splicers) .....	17
4.4.7 Save Screen Brightness (for screens loaded by LED 4K sending cards) .....	18
4.5 MVR Operations .....	18
4.5.1 Open MVR Windows .....	18
4.5.2 Delete MVR Windows .....	19
4.5.3 Move MVR Windows .....	19
4.6 IPC Operations .....	20
4.6.1 PTZ Control .....	20
4.6.2 Add IPC Sources .....	20
4.6.3 Delete IPC Sources .....	21
4.6.4 Add IPC Mosaic Sources .....	21
4.6.5 Delete IPC Mosaic Sources .....	22
A Serial Port Connections .....	23
B Configure JSON Format for IPC Mosaic Sources .....	24
B.1 Configure JSON Format for IPC Mosaic Sources .....	24

## 1. Communication Method

The H series video wall splicers communicate with the control PC via either of the following two methods.

### 1.1 UDP Communication

The video wall splicer is connected to the PC via Ethernet cable, and then the PC sends a UDP message to port 6000 on the video wall splicer. Each command is sent in a UDP packet.

### 1.2 Serial Communication

The video wall splicer is connected to the PC via RS232 serial cable.

- a) Baud rate: 9600
- b) Stop bits: 1
- c) Parity: None

You can also change the serial port parameters on the device front panel or Web page.

The control method includes a request and a response. One request packet corresponds to only one response packet to form a closed-loop communication. The previous two communication methods can work simultaneously; however the video wall splicer can execute only one command at a time and other commands will be ignored.

The video wall splicer provides an RS232 port for sending commands to its connected device and thus controls the connected device.

## 2. Command Format

The communication between the PC and video wall splicer uses the bidirectional protocol. All commands are in the following JSON format:

```
{
  "cmd": "****",
  "deviceId": 0,
  "param0": 0,
  "param1": 0
}
```

This instruction set uses the JSON data format.

All the instruction characters are case insensitive.

Each command starts with a "[" and ends with a "]".

All the parameters are separated by a ",".

Each "[" can contain several "{}" sub commands.

For example,

```
[
  {
    "cmd": "****",
  },
  {
    "cmd": "****",
  }
]
```

The PC sends the commands to the video wall splicer all the time, and the splicer responds in the following four ways:

1. Command executed successfully. If the command is a one-way command, the splicer returns the following message:

```
{
  "cmd": "****",
  "deviceId": 0,
  "ack": "Ok"
}
```

2. Command executed successfully. If the command is to read information from the splicer, the splicer returns the following message:

```
{
  "cmd": "****",
  "ack": "Ok",
  "deviceId": 0,
  "param0": 0,
  "param1": 0
}
```

3. If the command format is incorrect or is not supported due to compatibility issues, the splicer returns the following message:

```
{
  "cmd": "****",
  "deviceId": 0,
  "ack": "Error"
}
```

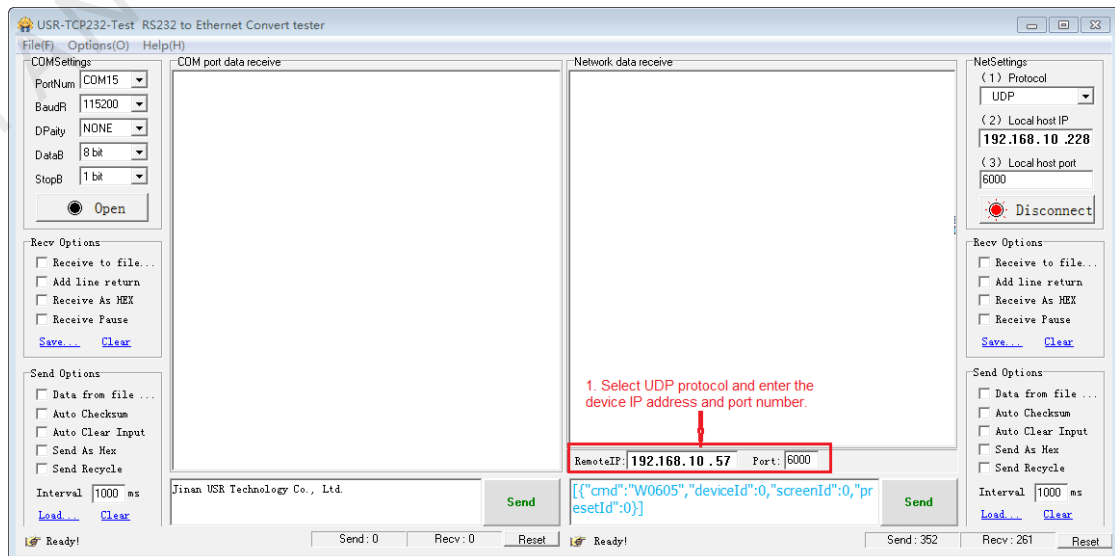
4. Command timeout.

When the splicer executes a command, other commands will be ignored and the PC may not receive any reply message. A one-second timeout for checking can be set on the PC after the first command is sent.

**Example: Load the first preset of the first screen**

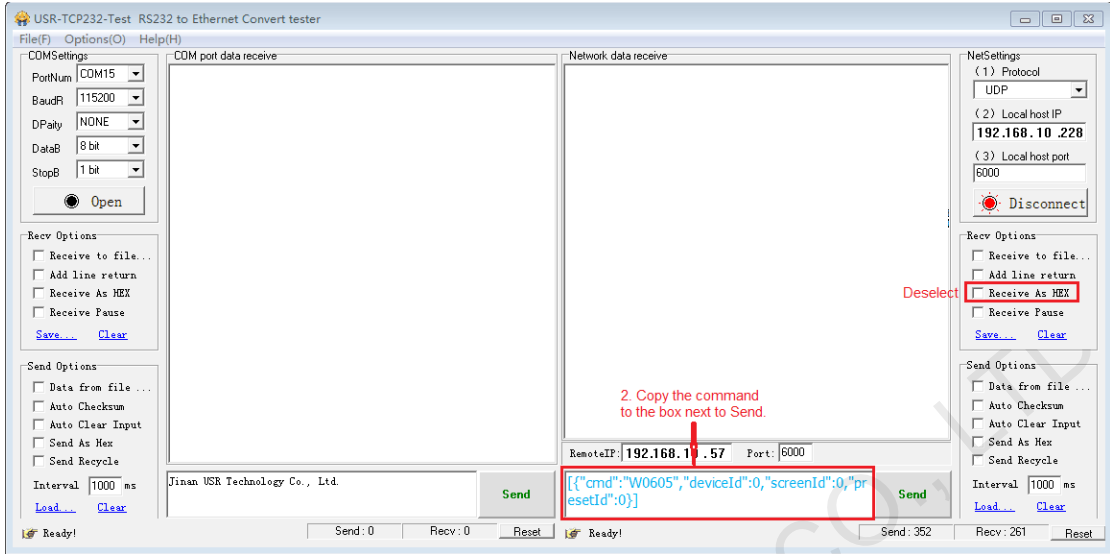
**Test tool:** 🧰 USR-TCP232-Test

1. Set the network or serial port parameters.

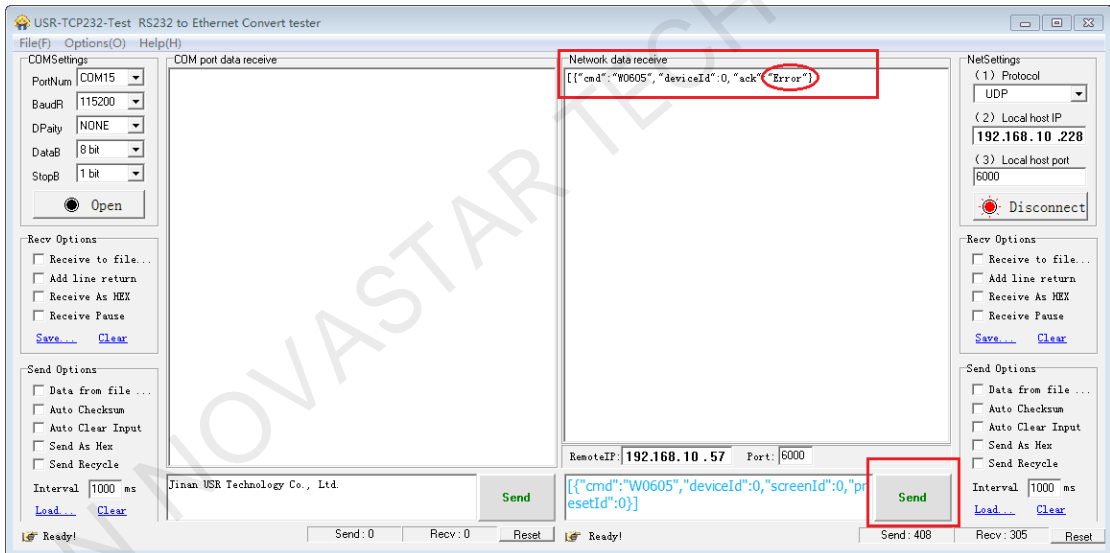


2. Copy the command to the text box next to **Send**.

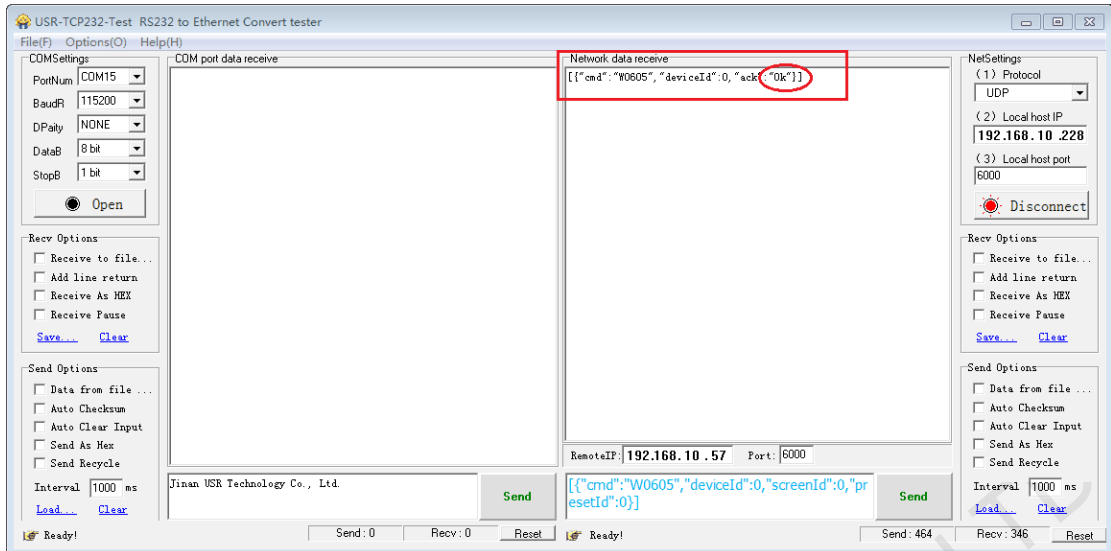
The command is as follows [{"cmd":"W0605","deviceId":0,"screenId":0,"presetId":0}]  
"screenId":0 represents the first screen and "presetId":0 represents the first preset.



3. Click **Send**.



The device will respond if the device recognizes the command. If the screen id or preset id does not exist, "ack":" Error" will be returned.



If the parameter is correct, "ack":" Ok" will be returned, indicating the success control over the device.

### 3. Parameter Descriptions

In this control protocol, the following parameters need to be explained.

1. inputId: ID of the main channel of the signal source
2. cropId: ID of the cropping channel of the signal source
3. screenId: Screen ID
4. outputId: Output connector ID
5. layerId: Layer ID
6. presetId: Preset ID

### 4. Command Protocols

#### 4.1 Layer Operations

The layer operations include opening, closing, moving, scaling, bringing to front, sending to back, switching input source, cropping and flipping.

##### 4.1.1 Open Layers

JSON Format	Description	Default
<pre>{   "cmd": "W0502",   "screenId": 0,   "deviceId": 0,   "general": {     "layerId": 0,     "name": "Layer0",     "sizeType": 0,     "type": 1,     "isBackground": false, </pre>	<p>Function: Add a specified layer to a screen.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"screenId": Screen ID</li> <li>"deviceId": Device ID</li> <li>"name": Layer name</li> <li>"layerId": Layer ID</li> <li>"sizeType" is described as follows:</li> <li>0: SL</li> </ul>	<p>When "cropId" is 255, the original source is used.</p> <p>"layerId ":</p> <p>If the message does not contain this field, the id will be assigned by the middleware automatically.</p>

<pre>"isFreeze":false, "flipType":0 }, "source":{ "sourceType":1, "inputId":0, "interfaceType":1, "cropId":255 }, "window":{ "width":1000, "height":1000, "x":1005, "y":1005 } }]</pre>	<p>1: DL 2: 4K</p> <p>"type" (layer type) is described as follows: 0: mix 1: SL</p> <p>"source": Layer source " sourceType " (layer source type) is described as follows: 0: No source 1: Input type " inputId ": Source ID "cropId": Crop source ID "interfaceType" (connector type) is described as follows: 1: EXP 2: Single Link DVI 3: Dual Link DVI 4: HDMI1.3 5: HDMI1.4 6: HDMI2.0 7: DP1.1 8: DP1.2 9: 3G-SDI 10: VGA 11: CVBS 12: YPbPr 13: RJ45 14: USB 15: HDBaseT 16: HDBaseT-4K 17: Optical fiber 18: 12G-SDI</p> <p>"window": Layer position "width", "height": Layer width and height "x", "y": Layer horizontal and vertical coordinates (The coordinates are absolute coordinates, not the relative coordinates to the screen. The default coordinates are 1005,1005 when configuring a screen on the Web)</p>	
---	--	--

#### 4.1.2 Close Layers

JSON Format	Description	Default
<pre>{ "cmd":"W0503",</pre>	<p>Function: Delete a specified layer from the screen.</p>	



<pre>"screenId":0, "deviceId":0, "layerId":0 }]</pre>	<p>Parameters:</p> <p>"screenId": Screen ID</p> <p>"deviceId": Device ID</p> <p>"layerId": Layer ID</p>	
---	---	--

#### 4.1.3 Adjust Layer Position and Size

JSON Format	Description	Default
<pre>{ "cmd":"W0505", "screenId":0, "deviceId":0, "layerId":0, "width":100, "height":100, "x":100, "y":100 }]</pre>	<p>Function: Read and write the position and size of a specified layer.</p> <p>Parameters:</p> <p>"deviceId": Device ID</p> <p>"screenId": Screen ID</p> <p>"layerId": Layer ID</p> <p>"width", "height": Layer width and height</p> <p>"x", "y": Layer horizontal and vertical coordinates</p>	

#### 4.1.4 Bring to Front / Send to Back

JSON Format	Description	Default
<pre>{ "cmd":"W0508", "screenId":0, "deviceId":0, "layerId":0, "layersZOrderAct":0 }]</pre>	<p>Function: Adjust the Z order of a specified layer on a specified screen.</p> <p>Parameters:</p> <p>"screenId": Screen ID</p> <p>"deviceId": Device ID</p> <p>"layerId": Layer ID</p> <p>"layersZOrderAct": Z order is described as follows:</p> <p>0: None</p> <p>1: Up</p> <p>2: Down</p> <p>3: Top</p> <p>4: Bottom</p>	

#### 4.1.5 Flip Layers

JSON Format	Description	Default
<pre>{ "cmd":"W0504", "layerId":1, "screenId":1, "deviceId":0, "flipType":0 }]</pre>	<p>Function: Get the basic information of a specified layer.</p> <p>Parameters:</p> <p>"deviceId": Device ID</p> <p>"screenId": Screen ID</p> <p>"layerId": Layer ID</p> <p>"flipType":</p> <p>0: Do not flip</p>	

	1: Flip horizontally 2: Flip vertically 3: Flip horizontally and vertically	
--	---	--

#### 4.1.6 Switch Layer Input Sources

JSON Format	Description	Default
<pre>{   "cmd": "W0506",   "deviceId": 0,   "screenId": 0,   "layerId": 0,   "inputId": 0,   "interfaceType": 1,   "cropId": 255 }</pre>	Function: Switch the layer source. Parameters: "deviceId": Device ID "screenId": Screen ID "layerId": Layer ID "inputId": Layer input source ID "interfaceType": Input source connector type "cropId": Crop source ID	When "cropId" is 255, the original source (not cropped source) will be used.

#### 4.1.7 Crop Layer Input Sources

JSON Format	Description	Default
<pre>{   "cmd": "W0207",   "inputId": 0,   "deviceId": 0,   "cropId": 0,   "name": "Input0-crop0",   "width": 500,   "height": 500,   "x": 0,   "y": 0 }</pre>	Function: Read and write the cropping information of a specified input. Parameters: "deviceId": Device ID "inputId": Layer input source ID "cropId": Crop source ID "width", "height": Crop width and height "name": Crop source name "x", "y": Horizontal and vertical coordinates of the cropped area	

## 4.2 Preset Operations

The preset operations include saving, clearing, loading and playback.

### 4.2.1 Load Presets

JSON Format	Description	Default
<pre>{   "cmd": "W0605",   "deviceId": 0,   "screenId": 0,   "presetId": 0 }</pre>	Function: Load a specified preset. Parameters: "deviceId": Device ID "screenId": Screen ID "presetId": Preset ID	

### 4.2.2 Get Preset Enum

Send instruction

JSON Format	Description	Default
<pre> {{ "cmd": "R0600", "param0": 0, "param1": 0 }}                     </pre>	Function: Get the device details. Parameters: "param0": Device ID, 0 by default "param1": Screen ID	

Response

JSON 格式	Description	Default
<pre> {{ "deviceId": 0, "screenId": 0, "presets": [   {     "name": "preset1",     "presetId": 0   },   {     "name": " preset2",     " presetId ": 1   } ] }}                     </pre>	Function: Describe the screen preset enum. Parameters: "deviceId": Device ID "presets": Preset ID list	

### 4.2.3 Clear Presets

Name	JSON Format	Attribute	Description	Default
	<pre> {{ "cmd": "W0603", "presetId": 0, "screenId": 0, "deviceId": 0 }}                     </pre>	W	Function: Clear a specified preset. Parameters: "deviceId": Device ID "screenId": Screen ID "presetId": Preset ID	

### 4.2.4 Save Presets

Name	JSON Format	Attribute	Description	Default
	<pre> {{ "cmd": "W0602", "screenId": 0, "deviceId": 0, "presetId": 0, "name": "preset1" }}                     </pre>	W	Function: Save a preset. Parameters: "deviceId": Device ID "name": Preset name "screenId": Screen ID "presetId": Preset ID	

	}]			
--	----	--	--	--

### 4.3 Device Operations

The device operations include:

- ✓ Read the statuses of fans, power supplies, temperature, voltage, card and Genlock.
- ✓ Set test patterns.

#### 4.3.1 Get Device Details

Send instruction

JSON Format	Description	Default
<pre>{   "cmd": "R0100",   "param0": 0 }</pre>	Function: Get the device details. Parameters: " param0", Device ID, 0 by default	

Response

JSON Format	Description	Default
<pre>{   "deviceId": 0,   "modelId": 29953,   "name": "",   "status": 0,   "protoVersion": "1.0.0.0",   "MAC": "00-23-5A-15-99-42",   "volt": 0,   "temp": 0,   "extSignal": 0,   "dataVersion": 0,   "slotList": {     "slotId": 0,     "modelId": 29953,     "cardType": 1,     "interfaces": {       "interfaceId": 1,       "interfaceType": 4,       "iSignal": 0,       "isUsed": 0,       "functionType": 1     }   },   "linkstatus": {     "link0": 0,     "link1": 0,     "link10": 0,     "link11": 0,     "link12": 0, </pre>	Function: Describe the basic information of a specified device. Parameters: "name": Device name "deviceId": Device ID "status": Device ready signal. 0: Device is busy. 1: Device is ready. "protoVersion": Protocol version "slotList": Card slot list (see protocol 3.2 and protocol 3.3) "powerList": Power supply (see protocol 0x0104) "genlock": Genlock information (see protocol 0x0105) "network": Network information, including IP address, subnet mask and gateway "fanList": Fan status 0: Normal 1: Abnormal "volt": Voltage status 0: Normal 1: Abnormal "temp": Temperature 0: Normal 1: Abnormal "extSignal": External Genlock signal	

<pre>"link13": 0, "link14": 0, "link15": 0, "link2": 0, "link3": 0, "link4": 1, "link5": 0, "link6": 0, "link7": 0, "link8": 0, "link9": 0   }, "lightstatus":{ "link0": 0, "link1": 0, },   }, "powerList":[{ "powerId": 1, "iSignal": 0 }], "fanList":[{ "fanId": 1, "status": 0 }], "genlock":{}, "network":{}, "usbStatus":{ "usbId": 0,   "status": 0 } }]</pre>	<pre>0: No signal 1: Signal accessed "dataVersion": Data version of this node "usbStatus": Control card USB status 0: No signal 1: Signal accessed</pre>	
---	--	--

### 4.3.2 Get Slot Information

Send instruction

JSON Format	Description	Default
<pre>{ "cmd":"R0102", "param0":0, "param1":0 }</pre>	<pre>Function: Get the device slot details. Parameters: " param0": Device ID, 0 by default "param1": Slot number, starting from 0</pre>	

Response

JSON Format	Description	Default
<pre>{</pre>	<pre>Function: Get the device slot details.</pre>	

<pre> "slotId":0, "deviceId" : 0, "status" : 1, //0 异常 , 1 正常 "modelId":29953, "cardType":4, "resolution": {     "height": 3000,     "refresh": 5000,     "width": 3840 }, "interfaces":[     { "interfaceId ":1, "interfaceType":4, "iSignal":0, "functionType": 1     } ], "linkstatus": { "link0": 0, "link1": 0, "link10": 0, "link11": 0, "link12": 0, "link13": 0, "link14": 0, "link15": 0, "link2": 0, "link3": 0, "link4": 1, "link5": 0, "link6": 0, "link7": 0, "link8": 0, "link9": 0 }, "lightstatus":{ "link0": 0, "link1": 0, }, "SenderColorDepth":0, "network": {     "mode": 0, //0 冗余模式,1 独立模式     "wanId": 0, //冗余模式下忽略此参数 </pre>	<p>Parameters:</p> <p>"slotId": Card slot ID</p> <p>"modelId": Model ID of the card inserted into the slot</p> <p>"status": 0: Abnormal 1: Normal</p> <p>"cardType": Card type [ 0: No card inserted 1: Input card slot 2: Output card slot 3: Sending card slot 4: MVR card slot ];</p> <p>"interfaces": List of the connectors of the card inserted into the slot (see protocol 0x0103)</p> <p>"linkstatus": LED 4K sending card status. Link 0-15 represents 16 Ethernet ports. 0: Ethernet cable not connected 1: Ethernet cable connected 2: Backup between Ethernet ports not enabled 3: Backup enabled</p> <p>lightstatus: Status of OPT port on LED 4K sending card 0: Not connected 1: Connected</p> <p>"SenderColorDepth": Bit depth of LED 4K sending card [0: 8bits, 1: 10bits, 2: 12bits, 3: 16bits]</p> <p>network: Network settings</p> <p>"mode": 0: Backup mode 1: Independent mode, available when CardType is MVR</p> <p>"wanId": This parameter can be ignored in 0 backup mode.</p> <p>"dhcp": 0: DHCP disabled, 1: DHCP enabled</p> <p>encoding : Encoding settings, available when CardType is MVR</p> <p>modelId, cardType and interfaces are read-only fields.</p>	
---	---	--

<pre>"dhcp":0, //DHCP 功能 "ips":[{"id": 0, "ip": { "ip0": 192, "ip1": 168, "ip2": 0, "ip3": 10 }, "subnetMask": { "subnetMask0": 255, "subnetMask1": 255, "subnetMask2": 255, "subnetMask3": 0 }, "gateway": { "gateway0": 192, "gateway1": 168, "gateway2": 0, "gateway3": 1 }, "mac": { "mac0": 00, "mac1": 01, "mac2": 02, "mac3": 03, "mac4": 04, "mac5": 05, } }, { "id": 1, "ip": { "ip0": 192, "ip1": 168, "ip2": 0, "ip3": 10 }, "subnetMask": { "subnetMask0": 255, "subnetMask1": 255, "subnetMask2": 255, "subnetMask3": 0 }, "gateway": { "gateway0": 192, "gateway1": 168, "gateway2": 0,</pre>		
---	--	--

<pre>"gateway3": 1 }, "mac": { "mac0": 00, "mac1": 01, "mac2": 02, "mac3": 03, "mac4": 04, "mac5": 05,     } } ] }, "encoding": { //编码     "mvrUrl": "xxx",     "echoUrl": "xxx" } }]</pre>		
---	--	--

### 4.3.3 Get Connector Information

Send instruction

JSON Format	Description	Default
<pre>[[ "cmd":"R0103", "param0":0, "param1":0, "param2":0 ]]</pre>	<p>Function: Get the slot connector information.</p> <p>Parameters:</p> <p>"param0": Device ID, 0 by default</p> <p>"param1": Slot number, starting from 0</p> <p>"param2": Connector number ranges from 0 to 3.</p>	

Response

JSON Format	Description	Default
<pre>[[ "interfaceId ": 1, "deviceId":0, "slotId":1, "interfaceType": 4, "iSignal": 0, "functionType": 1 ]]</pre>	<p>Function: Describe the slot connector.</p> <p>Parameters:</p> <p>"deviceId": Device ID</p> <p>"slotId": Slot ID</p> <p>"interfaceId ": Connector ID</p> <p>"interfaceType": Connector has the following types:</p> <p>[</p> <p>1: EXP</p> <p>2: Single Link DVI</p> <p>3: Dual Link DVI</p> <p>4: HDMI 1.3;</p> <p>5: HDMI 1.4</p>	



	6: HDMI 2.0 7: DP 1.1 8: DP 1.2 9: 3G-SDI 10: VGA 11: CVBS 12: YPbPr 13: RJ45 14: USB 15: HDBaseT 16: HDBaseT-4K 17: Optical fiber 18: 12G-SDI  ] ; "iSignal"; Signal status [For IN 0: No signal source 1: Signal source connected 2: Signal source disconnected (will not be kept after power off) For OUT 0: Not connected 1: Connected]	
--	--	--

#### 4.3.4 Reset to Factory Settings

JSON Format	Description	Default
<pre>{   "cmd": "W010B",   "deviceId": 0,   "type": 0 }</pre>	<p>Function: Reset the device to factory settings.</p> <p>Parameters:</p> <p>"type": Factory reset has the following types:</p> <pre>[   0: Reset, but keep IP   1: Reset, but keep IP and EDID   2: Reset, but keep IP, EDID and ID   3: Reset all ]</pre>	

## 4.4 Screen Operations

The screen operations include:

- ✓ Reading the screen configuration
- ✓ Setting BKG, OSD and brightness

#### 4.4.1 Adjust Screen Brightness (for screens loaded by LED 4K sending cards)

JSON Format	Description	Default
<pre>{   "cmd": "W0410",   "deviceId": 0,   "screenId": 0,   "brightness": 0 }</pre>	Function: Adjust the screen brightness. Parameters: "deviceId": Device ID "screenId": Screen ID "brightness": Screen brightness value (0–100)	

#### 4.4.2 Set Screen FTB

JSON Format	Description	Default
<pre>{   "cmd": "W0409",   "deviceId": 0,   "screenId": 0,   "type": 0 }</pre>	Function: Set whether to make the screen fade to black. Parameters: "deviceId": Device ID "screenId": Screen ID "type": <b>[0: FTB enabled, 1: FTB disabled]</b>	

#### 4.4.3 Get Screen Output Information

Send instruction

JSON Format	Description	Default
<pre>{   "cmd": "R0405",   "param0": 0,   "param1": 0 }</pre>	Function: Get the slot connector details. Functions: "param0": Device ID, 0 by default "param1": Screen ID, starting from 0	

Response

JSON Format	Description	Default
<pre>[   {     "screenId": 0,     "deviceId": 0,     "mosaic": {       "row": 1,       "column": 2     },     "size": {       "width": 1920,       "height": 1080     }   } ]</pre>	Function: Describe the output connector mode of a specified screen. Parameters: "deviceId": Device ID "screenId": Screen ID "screenInterfaces ": List of connectors of a screen "interfaceld": Assigned connector ID after the screen is configured. This ID is unique and read-only. "outputId": Output ID	

<pre> },   "screenInterfaces": [     {       "interfaceld": 0,       "outputld": 1,       "x": 0,       "y": 0,       "cropx": 0,       "cropy": 0,       "width": 1920,       "height": 1080,       "resolution": {         "width": 1920,         "height": 1080,         "refresh": 6000       }     }   ] } ] </pre>	<p>id=255: Output connector not assigned</p>	
--	--	--

#### 4.4.4 Screen Layer Enum

Send instruction

JSON Format	Description	Default
<pre> {{   "cmd": "R0500",   "param0": 0,   "param1": 0 }} </pre>	<p>Function: Get the slot connector details.</p> <p>Parameters:</p> <p>"param0": Device ID, 0 by default</p> <p>"param1": Screen ID, starting from 0</p>	

Response

JSON Format	Description	Default
<pre> {{   "screenld": 0,   "deviceld": 0,   "screenLayers": [     {       "name": "layer1",       "layerld": 0     },     {       "name": "layer2",       "layerld": 1     }   ] }} </pre>	<p>Function: Describe the layer enum of a specified screen.</p> <p>Parameters:</p> <p>"screenLayers": Layer list</p> <p>"name": Layer name</p> <p>"layerld": Layer ID</p>	

}]		
----	--	--

#### 4.4.5 Screen Color

JSON Format	Description	Default
<pre> {{   "cmd": "W040E",     "deviceId": 0,     "screenId": 0,     "imageQualityMode": 0,     "eyeCare": 0,     "contrast": {       "all": 50,       "R": 50,       "G": 50,       "B": 50     },     "brightness": {       "all": 50,       "R": 50,       "G": 50,       "B": 50     },     "hue": 0,     "saturation": 50,     "colorTemperature": 6500   }} </pre>	<p>Function: Write the contrast, brightness, saturation and hue of a specified screen.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"screenId": Screen ID</li> <li>"deviceId": Device ID</li> <li>"contrast": Contrast (RGB component value is the same as the overall value)</li> <li>"brightness": Brightness (RGB component value is the same as the overall value)</li> <li>"hue": Hue</li> <li>"saturation": Saturation</li> <li>"all": Overall value of brightness or contrast</li> <li>"R", "G" and "B" represent the component values of the brightness or contrast.</li> <li>"imageQualityMode": Display mode</li> <li>0: Standard</li> <li>1: Document</li> <li>2: Conference</li> <li>3: Video</li> <li>"eyeCare": Eye saver mode</li> <li>0: Off</li> <li>1: On</li> <li>"colorTemperature": The color temperature ranges from 2000 to 10000.</li> </ul> <p>Because the brightness, saturation, contrast, hue and color temperature are all required in CSC calculation, it is recommended to send all the data uniformly to ensure data consistency.</p>	

#### 4.4.6 Screen 3D (for screens loaded by LED 4K sending cards and for V1.3.0.0 or later version of the H series video wall splicers)

JSON Format	Description	Default
<pre> {{   "cmd": "W0415",   "screenId": 1,   "deviceId": 0, </pre>	<p>Function: Set the 3D mode parameters.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> </ul>	

<pre>"enable":0, "type":0, " LeftFirstThenRight":0, "startPointUpdown":0, "startPointLeftright":0, "Emitter ":0, "delayTimeMs":0, "delayTimeUs":0 }]</pre>	<pre>" screenId": Screen ID " enable " : 0: Turn off 3D mode. 1: Turn on 3D mode. " Emitter " : 0: Turn off the emitter. 1: Turn on the emitter. " startPointUpdown " : Top-and- bottom, right eye start startPointLeftright " : Side-by- side, right eye start "delayTimeMs": Delay time (unit: ms) "delayTimeUs": Delay time (unit: us)</pre>	
--	---	--

#### 4.4.7 Save Screen Brightness (for screens loaded by LED 4K sending cards)

JSON Format	Description	Default
<pre>{ "cmd":"W0417", "deviceId":0, "screenId":0, "brightness":0 }</pre>	<p>Function: Save the screen brightness.</p> <p>Parameters:</p> <p>"deviceId": Device ID</p> <p>" screenId": Screen ID</p> <p>"brightness": Screen brightness, ranging from 0 to 100</p>	

## 4.5 MVR Operations

The MVR operations include opening MVR windows, switching window sources, and setting MVR window border and UMD.

### 4.5.1 Open MVR Windows

JSON Format	Description	Default
<pre>{ "cmd":"W0802", "deviceId": 0, "MVRId": 1, &gt;windowId": 0, "slotId": 0, "srcId": 0, &gt;windowType": 0, &gt;window": { "width": 100, "height": 100, "x": 0, "y": 0 }, }</pre>	<p>Function: Open an MVR window.</p> <p>Parameters:</p> <p>"deviceId": Device ID</p> <p>"MVRId": MVR ID</p> <p>"umdName": Window name</p> <p>"windowId": Window ID</p> <p>"srcId": Window source ID</p> <p>"windowType": Window type</p> <p>[</p> <p>0: Input</p> <p>1: PGM</p> <p>2: PVW</p> <p>3: IPC</p>	<p>"MVRId ": If the message does not contain this field, the id will be assigned by the middleware automatically, and the response packet must contain the id.</p> <p>The response packet must be in the following format:</p> <pre>[[{"id" : 0}]</pre>

<pre>"umd": {   "umdName": "umd",   "umdEnable": 0,   "fontColor": {     "R": 100,     "G": 100,     "B": 100   },   "border": {     "borderWidth": 5,     "color": {       "R": 100,       "G": 100,       "B": 100     }   } }</pre>	<pre>]</pre>	
--	--------------	--

#### 4.5.2 Delete MVR Windows

JSON Format	Description	Default
<pre>{   "cmd": "W0803",   "deviceId": 0,   "MVRId": 1,   "windowId": 0, }</pre>	<p>Function: Delete a specified MVR window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"MVRId": MVR ID</li> <li>"windowId": Window ID</li> </ul>	

#### 4.5.3 Move MVR Windows

JSON Format	Description	Default
<pre>{   "cmd": "W0804",   "deviceId": 0,   "MVRId": 1,   "windowId": 1,   "srcId": 1,   "windowType": 0,   "window": {     "width": 100,     "height": 100,     "x": 0,     "y": 0   } }</pre>	<p>Function: Read and write the basic information of the MVR window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"MVRId": MVR ID</li> <li>"windowId": Window ID</li> <li>"MVRId": MVR ID</li> <li>"srcId": Window source ID</li> <li>"windowType": Window type</li> </ul> <p>[ 0: Input 1: PGM 3: PWW ];</p>	

	"window": Window size	
--	-----------------------	--

## 4.6 IPC Operations

### 4.6.1 PTZ Control

JSON Format	Description	Default
<pre>{   "cmd": "W0A07",   "deviceId": 0,   "slotId": 0,   "ipcSourceId": 0,   "controlType": 0,   "speed": 0 }</pre>	<p>Function: Set the IPC source position.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"slotId": Card slot number</li> <li>"ipcSourceId": IPC source number</li> <li>"controlType":           <ul style="list-style-type: none"> <li>0: Up</li> <li>1: Down</li> <li>2: Left</li> <li>3: Right</li> <li>4: Zoom in</li> <li>5: Zoom out</li> <li>6: Stop</li> </ul> </li> <li>"speed": Reserved</li> </ul>	

### 4.6.2 Add IPC Sources

JSON Format	Description	Default
<pre>{   "cmd": "W0A04",   "deviceId": 0,   "slotId": 0,   "ipcSourceId": 0,   "name": "xxx",   "protocol": {     "type": 0,     "gb28181": {       "platformId": "xx",       "platformPort": 7100,       "platformIp": "192.168.0.100",       "localId": "xx",       "localPort": 5060,       "localIp": "192.168.0.101",       "sipId": "xx",       "sipPort": 5500,     },     "rtsp": {       "rtspUrl": "xxx",       "rtspUserName": "xx",     }   } }</pre>	<p>Function: Add an IPC input source.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"slotId": Card slot number</li> <li>"ipcSourceId": IPC source number</li> <li>"name": IPC source name</li> <li>"type": Protocol type           <ul style="list-style-type: none"> <li>0: GB28181</li> <li>1: RTSP</li> <li>2: ONVIF</li> </ul> </li> </ul>	

<pre>"rtspPassWord": "***", }, "onvif": { "onvifType": 0, "onvifIp": "192.168.0.101", "onvifUserName": "xx", "onvifPassWord": "xx", } } }]</pre>		
--	--	--

### 4.6.3 Delete IPC Sources

JSON Format	Description	Default
<pre>{ "cmd": "W0A06", "deviceId": 0, "sources": [ "ipcSourceList": [ "ipcSourceId": 0 ], "slotId": 0 } }</pre>	<p>Function: Delete an IPC input source.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"slotId ": Card slot number</li> <li>"ipcSourceList": IPC source list</li> <li>"ipcSourceId": IPC source number</li> </ul>	

### 4.6.4 Add IPC Mosaic Sources

JSON Format	Description	Default
<pre>{ "cmd": "W0A05", "deviceId": 0, "slotId": 0, "inputId": 0, "modeName": "xx", "mosaic": { "row": 1, "column": 1 }, "sourcePosList": [ "ipcSourceId": 0, "x": 0, "y": 0, "width": 3840, "height": 2160 } }</pre>	<p>Function: Configure the IPC input card working mode, mosaic mode and layout.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>"deviceId": Device ID</li> <li>"slotId ": Card slot ID</li> <li>"modeName": Current mode</li> <li>"mosaic": Current mosaic layout</li> <li>"sourcePosList ": IPC source position list under mosaic mode</li> <li>"ipcSourceId ": IPC source ID</li> <li>"x": Horizontal coordinate</li> <li>"y": Vertical coordinate</li> <li>"width": Width</li> <li>"height": Height</li> </ul>	<p>For JSON format for adding multiple mosaic sources, please see Appendix B.</p>

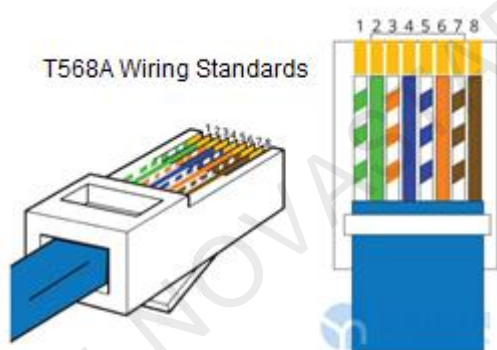
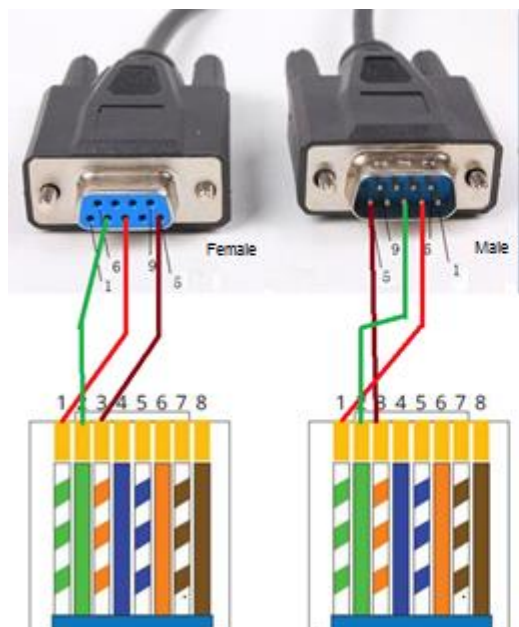


#### 4.6.5 Delete IPC Mosaic Sources

JSON Format	Description	Default
<pre> [[   "cmd":"W0A02",   "deviceId":0,   "slotId": 0,   "modeName":"xx",   "mosaic": {     "row": 0,     "column": 0   } ]] </pre>	<p>Function: Delete an IPC mosaic source.</p> <p>Parameters:</p> <p>"deviceId": Device ID  "slotId ": Card slot ID  "modeName": Current mode  "mosaic": Current mosaic layout  "row": 0  "column": 0</p> <p>Indicates the IPC mosaic source will be deleted if the row and column are both set to 0.</p>	

XI'AN NOVASTAR TECH CO., LTD

# A Serial Port Connections



# B Configure JSON Format for IPC Mosaic Sources

## B.1 Configure JSON Format for IPC Mosaic Sources

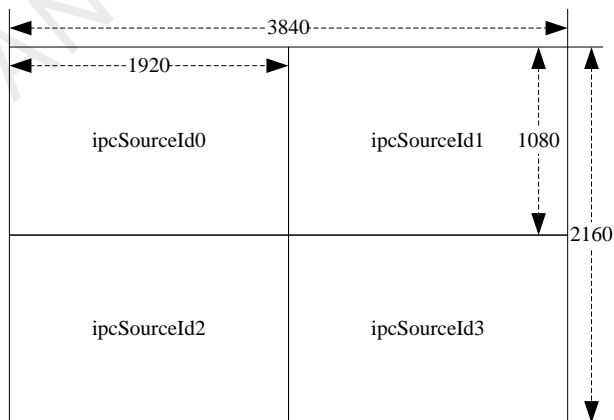
For a 3840x2160 IPC mosaic source using a row\*column layout,

- Single source width =  $3840/\text{row}$ , single source height =  $2160/\text{column}$
- inputid = first input connector ID (slotId\*4) of the current IPC card

For example, a mosaic source using a 1x1 layout

```
{  
  "cmd": "W0A05",  
  "deviceId": 0,  
  "slotId": 0,  
  "inputId": 0,  
  "modeName": "xx",  
  "mosaic": {  
    "row": 1,  
    "column": 1  
  },  
  "sourcePosList": [{  
    "ipcSourceId": 0,  
    "x": 0,  
    "y": 0,  
    "width": 3840,  
    "height": 2160  
  }]  
}
```

For example, a mosaic source using a 2x2 layout is shown as below:



```
{  
  "cmd": "W0A05",  
  "deviceId": 0,
```

```
"slotId": 0,  
"inputId": 0,  
"modeName": "xx",  
"mosaic": {  
  "row": 2,  
  "column": 2  
},  
"sourcePosList": [{  
  "ipcSourceId": 0,  
  "x": 0,  
  "y": 0,  
  "width": 1920,  
  "height": 1080  
},  
{  
  "ipcSourceId": 1,  
  "x": 1920,  
  "y": 0,  
  "width": 1920,  
  "height": 1080  
},  
{  
  "ipcSourceId": 2,  
  "x": 0,  
  "y": 1080,  
  "width": 1920,  
  "height": 1080  
},  
{  
  "ipcSourceId": 3,  
  "x": 1920,  
  "y": 1080,  
  "width": 1920,  
  "height": 1080  
}]  
}]
```

NOVA STAR TECH CO., LTD

**Copyright © 2021 Xi'an NovaStar Tech Co., Ltd. All Rights Reserved.**

No part of this document may be copied, reproduced, extracted or transmitted in any form or by any means without the prior written consent of Xi'an NovaStar Tech Co., Ltd.

**Trademark**

**NOVA STAR** is a trademark of Xi'an NovaStar Tech Co., Ltd.

**Statement**

Thank you for choosing NovaStar's product. This document is intended to help you understand and use the product. For accuracy and reliability, NovaStar may make improvements and/or changes to this document at any time and without notice. If you experience any problems in use or have any suggestions, please contact us via the contact information given in this document. We will do our best to solve any issues, as well as evaluate and implement any suggestions.

XI'AN NOVASTAR TECH CO., LTD

| [Official website](http://www.novastar.tech)  
| [www.novastar.tech](http://www.novastar.tech)

| [Technical support](mailto:support@novastar.tech)  
| [support@novastar.tech](mailto:support@novastar.tech)