
JSON RPC for Event Master Processors v9.2

General

JSON (JavaScript Object Notation) is a lightweight format that is used for interchanging data. It is based on a subset of JavaScript language: the way objects are built in JavaScript.

Introduction to JSON

JSON is built on two structures:

- A collection of name/value pairs: In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values: In most languages, this is realized as an array, vector, list, or sequence.

Here is an example of JSON data:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    "212 555-1234",
    "646 555-4567"
  ]
}
```

How JSON interacts with Event Master processors

JSON uses JSON RPC (REST based) to interact with the Event Master processors such as E2, S3-4K, and EX. JSON-RPC is a remote procedure call protocol encoded in JSON.

JSON-RPC works by sending a request to a server implementing this protocol. The client in that case is typically software intending to call a single method of a remote system. Multiple input parameters can be passed to the remote method as an array or object, whereas the method itself can return multiple output data as well.

There are JSON RPCs defined to perform tasks on the Event Master system. The user needs to send JSON requests through their application or open source application like Postman.

These applications should send requests on IP where the Event Master host processor is running and fixed port 9999 (Webserver of Event Master processor is running on port 9999).

Postman

Postman is an application that you can use to test the Event Master JSON API.

1. Go to the GetPostman website.
(<https://www.getpostman.com/>)
2. Download the free Postman app.
3. Install Postman on your machine.
4. Launch Postman and follow these steps to send JSON commands to the Event Master processor.
 - a. Select POST from the dropdown next to the URL text box.
 - b. Enter the request URL with port 9999.
 - c. Type "Content-Type" for header and "application/json" as value of this header.
(See <https://www.getpostman.com/docs/requests> for more information on sending requests.)

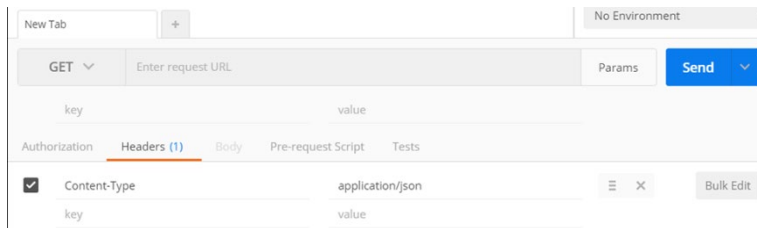


Image B-1
Content-Type and application/json

5. Select **Body** and click on raw from the buttons available below the URL text box.
6. Write the request in the body.

A request is a call to a specific method provided by a remote system. It must contain three certain properties:

- method—a String with the name of the method to be invoked.
 - params—an Object or Array of values to be passed as parameters to the defined method.
 - id—a value of any type used to match the response with the request to which it is replying.
- The receiver of the request must reply with a valid response to all received requests. A response must contain the properties mentioned below.

- result—the data returned by the invoked method. If an error occurred while invoking the method, this value must be null.
- error—a specified error code if there was an error invoking the method, otherwise null.
- id—the id of the request to which it is responding.

For example:

```
{ "params": {}, "method": "getFrameSettings", "id": "1234", "jsonrpc": "2.0" }
```

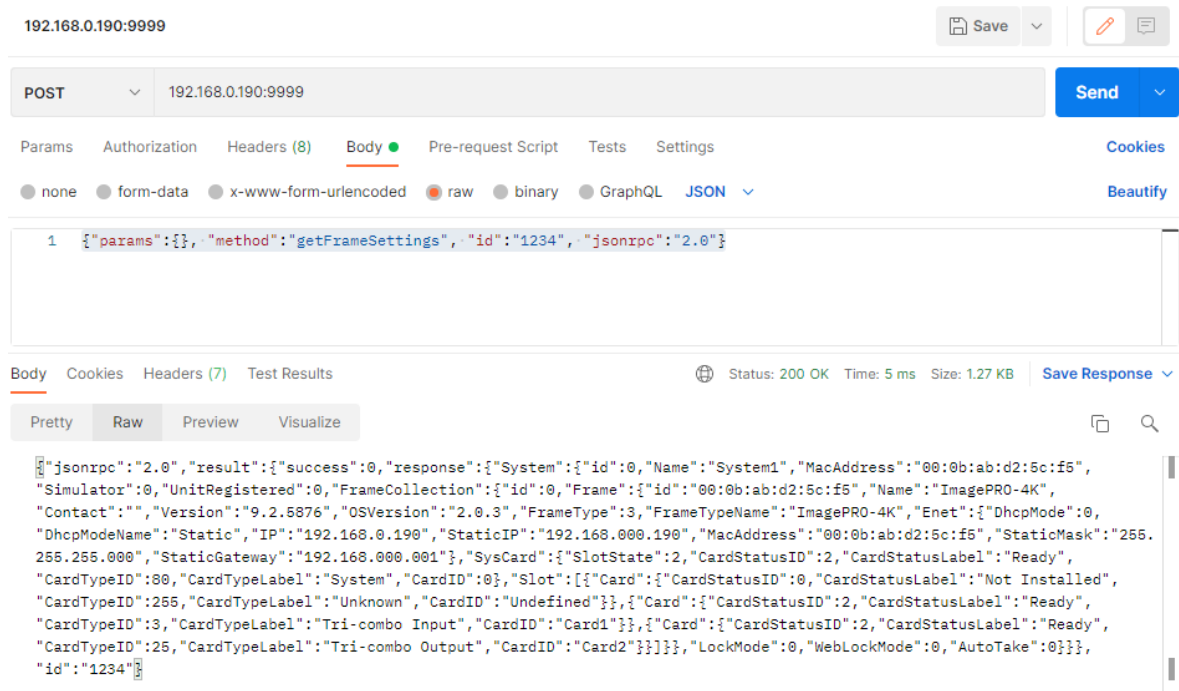
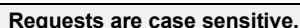


Image B-2
getFrameSettings API – example [powerStatus_API.png]

Here, **getFrameSettings** is the JSON API to be sent to an ImagePRO-4K video processor.

Here are some of the JSON APIs defined for controlling the Event Master processor. If you have a host + client(s) Event Master system, JSON must be sent to the host unit. For all the requests, this section explains the parameter passed or used in the RPC calls. If the “params” object is blank that means the API doesn’t require a parameter.



- **Definition:**
 - This API shows system information, including all the frames information.
- **Request:**
 - `params: {}`
- **Response:**
 - ```
{
 "jsonrpc": "2.0",
 "result": {
 "success": 0,
 "response": {
 "System": {
 "id": 0,
 "Name": "System1",
 "MacAddress": "00:13:95:31:ae:af",
 "Simulator": 0,
 "UnitRegistered": 0,
 "FrameCollection": {
 "id": 0,
 "Frame": {
 "id": "00:13:95:31:ae:af",
 "Name": "E2",
 "Contact": "",
 "Version": "9.1.5610",
 "OSVersion": "3.1.1",
 "FrameType": 0,
 "FrameTypeName": "E2",
 "Enet": {
 "DhcpMode": 0,
 "DhcpModeName": "Static",
 "IP": "192.168.0.175",
 "StaticIP": "192.168.000.175",
 "MacAddress": "00:13:95:31:ae:af",
 "StaticMask": "255.255.255.000",
 "StaticGateway": "192.168.000.001"
 },
 "SysCard": {
 "SlotState": 2,
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 80,
 "CardTypeLabel": "System",
 "CardID": 0,
 "Slot": [
 {
 "Card": {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 70,
 "CardTypeLabel": "Expansion",
 "CardID": "Card0"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 70,
 "CardTypeLabel": "Expansion",
 "CardID": "Card1"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 1,
 "CardTypeLabel": "SDI Input",
 "CardID": "Card2"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 4,
 "CardTypeLabel": "HDMI 2.0 Input",
 "CardID": "Card3"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 0,
 "CardTypeLabel": "DVI Input",
 "CardID": "Card4"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 2,
 "CardTypeLabel": "HDMI/DP Input",
 "CardID": "Card5"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 3,
 "CardTypeLabel": "Tri-combo Input",
 "CardID": "Card6"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 4,
 "CardTypeLabel": "HDMI 2.0 Input",
 "CardID": "Card7"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 4,
 "CardTypeLabel": "HDMI 2.0 Input",
 "CardID": "Card8"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 3,
 "CardTypeLabel": "Tri-combo Input",
 "CardID": "Card9"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 22,
 "CardTypeLabel": "HDMI 1.4 Output",
 "CardID": "Card10"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 26,
 "CardTypeLabel": "HDMI 2.0 Output",
 "CardID": "Card11"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 25,
 "CardTypeLabel": "Tri-combo Output",
 "CardID": "Card12"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 42,
 "CardTypeLabel": "MVR Gen2",
 "CardID": "Card13"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card14"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card15"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card16"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card17"
 },
 {
 "CardStatusID": 0,
 "CardStatusLabel": "Not Installed",
 "CardTypeID": 255,
 "CardTypeLabel": "Unknown",
 "CardID": "Undefined"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card19"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card20"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card21"
 },
 {
 "CardStatusID": 2,
 "CardStatusLabel": "Ready",
 "CardTypeID": 50,
 "CardTypeLabel": "VPU Scaler",
 "CardID": "Card22"
 }
]
 }
 }
 },
 "LockMode": 0,
 "WebLockMode": 0,
 "AutoTake": 0
 }
 },
 "id": "1234"
 }
}
```
  - System — System name and index.
  - FrameCollection — Collection of frames in a system containing frame information.
  - Frame — Contains frame information.
  - Id — Mac Id of the frame.
  - Name — Name of the frame.

- Contact — Contact information.
  - Version — Current version of the software installed on the frame.
  - OSVersion — Current OS version installed on the frame.
  - FrameType — 0: E2, 1:S3, 2: Ex, 3:IP-4K, 5:E2-Gen2, 6:PDS-4K
  - FrameTypeName — Type of the frame: E2, S3, Ex, etc.
  - Enet — Ethernet settings.
  - SysCard — System card information.
  - Slot — List of Input/Output/Expansion card information.
- success: (0=success, anything else is an error)
- Example:
  - {"params": {}, "method": "getFrameSettings", "id": "1234", "jsonrpc": "2.0"}

## **powerStatus**

- Definition:
  - This queries the power plug status of the Event Master processor. (There can be 1 or 2 power slots in Event Master processor).
- Request:
  - params: {} - It doesn't require any parameter.
- Response:
  - response: {FrameId1:{PwrStatus1, PwrStatus2},{FrameId2 :{ PwrStatus1, PwrStatus2}}
    - PwrStatus1 gives the power status of the 1st slot in Event Master processor with frame id FrameId1, FrameId2.
    - PwrStatus2 gives the power status of the 2nd slot in Event Master processor with frame id FrameId1, FrameId2.
    - 0: Power supply module is not present.
    - 1: Power supply module is present, but there is no power cable.
    - 2: Power supply module is present, and the cable is plugged in, but there is no DC current.
    - 3: Power supply module is present, and everything is OK.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {}, "method": "powerStatus", "id": "1234", "jsonrpc": "2.0"}

## **listInputs**

- Definition:
  - This queries the list of inputs configured.
- Request:
  - params: {"inputId": x},
    - "x" can be:
      - -1: For All inputs. (inputId is optional parameter, if not provided list of all inputs configured will be returned.)
      - Anything else will be treated as input config index and will return input of that index.
- Response:
  - response: Array of: [{"id":8, "Name": "DPBackground1", "SyncStatus": "None", "VideoStatus": "No Sync", "Format": "1920x1080p @59.94", "Color\_Space": "RGB, Full Range", "Colorimetry": "BT.709", "GammaFx": "SDR", "ColorDepth": "N/A", "Capacity": "SL"}]
    - Response contains the array of inputs. Above response contains id, name, Sync status, video status, Format, Color Space, Colorimetry, Gamma Fx, Color Depth and capacity.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"inputId": 1}, "method": "listInputs", "id": "1234", "jsonrpc": "2.0"}

### ***listOutputs***

- Definition:
  - This queries the list of outputs configured.
- Request:
  - params: {"outputCfgId": x},  
"x" can be:
    - -1: For All outputs. (outputCfgId is not mandatory parameter, if not provided list of all outputs configured will be returned.)
    - Anything else will be treated as output config index and will return output of that index.
- Response:
  - response: Array of: [{"id":2,"Name":"HDMIOutput3","Format":"1920x1080p@59.94","ColorSampleBit":"RGB/4:4:4/8","Color\_Space":"RGB, Full Range","Colorimetry":"BT.709","GammaFx":"SDR","HDRMetaFileIndex":0,"Capacity":"4K"}]
  - Response contains the array of outputs. Above response contains id, name, Format, Color/Sample/Bit, Color space, Colorimetry, Gamma Fx, HDRMetaFileIndex and capacity.
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"outputCfgId": 1}, "method":"listOutputs", "id":"1234", "jsonrpc":"2.0"}

### ***resetFrameSettings***

- Definition
  - Expose ALL reset types on Event Master processor with different options.
- Request:
  - params: {"reset":x},  
"x" can be 0 – 5
    - 0: Soft reset.
    - 1: Factory reset.
    - 2: Factory reset (save IP).
    - 3: Factory reset (save IP/EDID).
    - 4: Factory reset (save IP/EDID/VPID).
    - 5: Power Down.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example
  - {"params":{"reset": 0}, "method":"resetFrameSettings", "id":"1234", "jsonrpc":"2.0"}

### ***3dControl***

- Definition:
  - This API provides the option to modify 3d Controls.
- Request:
  - params: {"id": id, "type": x, "syncSource": y, "syncInvert": z}
    - id — Index of the input config.
    - type — "x" can be: 0 – Type Off. 0 is the default value for the type parameter. 1 – Type Sequential.
    - syncSource — "y" can be: 1 – mini-Din 1, 2 – mini-Din 2, 3 – mini-Din 3, 4 – mini-Din 4. Default value is 1.
    - syncInvert — "z" can be: 0 – Type Off. 0 is the default value for the syncInvert. 1 – Type Invert.
    - To reset, do not provide any parameter except "id".
- Response:
  - response: {"id": 0, "Name": "InSource1", "HSize": 3840, "VSize": 1080, "Src-Type": 0, "InputCfgIndex": -1, "StillIndex": 0, "DestIndex": -1, "UserKeyIndex": -1, "Mode3D": 0, "Freeze": 1, "Capacity": 2, "InputCfgVideoStatus": 4}
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 1, "type": 0, "syncSource": 1, "syncInvert": 0}, "method":"3dControl", "id":"1234", "jsonrpc":"2.0"}

### **3dControlOutput**

- Definition:
  - This API provides the option to modify 3d Controls on output configs.
- Request:
  - `params: {"outputId": 0, "3Dtype": 1}, "method":"3dControlOutput", "id":"1234", "jsonrpc":"2.0"}`
    - outputId – Index of the output config.
    - 3Dtype — "x" can be: 0 – Type Off. 0 is the default value for the type parameter. 1 – Type Sequential.
    - To reset, do not provide any parameter except "outputId".
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - `{"params": {"outputId": 0, "3Dtype": 1}, "method":"3dControlOutput", "id":"1234", "jsonrpc":"2.0"}`

### **listOperators**

- Definition:
  - This queries the list of outputs configured.
- Request:
  - `params: {}`
- Response:
  - `response: Array of [{"id":0,"Name":"Operator 1", "Enable":0, "StartRange":1, "EndRange":1000, "InvertColor":0,"DestCollection":[]},{ "id":1,"Name":"Operator 2", "Enable":1, "StartRange":3, "EndRange":4, "InvertColor":0, "DestCollection":[{"id":0, "DestType":1, "DestXmlId":0, "Name":"ScreenDest1"}, {"id":1, "DestType":0, "DestXmlId":0, "Name":"AuxDest1"}]},{ "id":2,"Name":"Operator 3", "Enable":0, "StartRange":1, "EndRange":1000, "InvertColor":0, "DestCollection":[{"id":0, "DestType":1, "DestXmlId":0, "Name":"ScreenDest1"}]}`
    - Response contains the array of multioperators. Above response contains id, name, Enable mode, Start Range of preset serial number, End Range of preset serial number, color of controller is inverted for this operator and collection of destination selected for this operator.
  - success: (0=success, anything else is an error)
- Example:
  - `{"params":{}, "method":"listOperators", "id":"1234", "jsonrpc":"2.0"}`

### **configureOperator**

- Definition:
  - This API helps user to configure operator.
- Request:
  - `params: {"operatorId": 2, "name": "qwert", "startRange":89, "endRange":95, "enable": 0, "add":{"destType": 1, "destIndex":0}, "remove" :{"destType": 1, "destIndex":1}}`
    - operatorId: Operator index which needs to be configured.
    - name: User can set the name of the operator.
    - startRange : This is start range of preset serial number assigned to the operator.
    - endRange: This is end range of preset serial number assigned to the operator.
    - enable: enable(1) or disable(0) the operator.
    - add/remove: Add/Remove destination for the operator.
      - destType: 0 – aux, 1 – screen, 2 – Super destination and 3 – super aux.
      - destIndex: destination index.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:

- {"params": {"operatorId": 2, "name": "qwert", "startRange":89, "endRange":95, "enable": 1}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}  
*// Enables Operator 3 while changing its name to qwert and assigns preset range 89-95.*
- {"params": {"operatorId": 1, "add":{"destType": 1, "destIndex":0}, "remove":{"destType": 1, "destIndex":1}}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}  
*// For Operator 2 – Removes Screen Destination 1 and adds Screen Destination 2 for this operator to control.*
- {"params": {"operatorId": 2, "name": "operator3", "endRange":134}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}  
*// For Operator 3 – Renames the profile to operator3 and changes the ending preset in its range to 134.*
- {"params": {"operatorId": 1, "name": "operator2", "startRange":21}, "method":"configureOperator", "id":"1234", "jsonrpc":"2.0"}  
*// For Operator 2 – Renames the profile to operator2 and changes the starting preset in its range to 21.*

## listSources

- Definition:
  - This API lists all the input sources with properties.
- Request:
  - params: {"type": x}
    - o 0 — Show all the input sources.
    - o 0 is the default value for the type parameter.
    - o 1 — Only background sources.
- Response:
  - response: Array of: {"id": 0, "Name": "InSource1", "HSize": 3840, "VSize": 1080, "SrcType": 0, "InputCfgIndex": -1, "StillIndex": 0, "DestIndex": -1, "UserKeyIndex": -1, "Mode3D": 0, "Freeze": 1, "Capacity": 2, "InputCfgVideoStatus": 4}
  - success: (0=success, anything else is an error)
  - Parameter to look for is "InputCfgVideoStatus". Possible values:
    - 0 = Invalid; there is sync, but cannot acquire / lock mismatch
    - 1 = Valid; Video is OK
    - 2 = MismatchFormat; Format mismatch between input cfg and connector(s)
    - 3 = OutOfRange; connector capacity is too low to acquire format
    - 4 = NoSync; no video
- Example:
  - {"params": {"type": 0}, "method":"listSources", "id":"1234", "jsonrpc":"2.0"}

## listStill

- Definition:
  - This API lists all the stills with properties such as id, Name, H/V size, LockMode, StillState, PngState, File size.
- Request:
  - params: {}
- Response:
  - response: Array of :
 

```
[{"id":0,"Name":"StillStore1","LockMode":0,"HSize":{"Min":0,"Max":99999,"$t":1920},"VSize":{"Min":0,"Max":99999,"$t":1080}, "StillState":{"Min":0,"Max":4,"$t":3}, "PngState":{"Min":0,"Max":2,"$t":0},"FileSize":{"Min":0,"Max":100000,"$t":9331.2}]
```

    - o id — Index of still store.
    - o Name — Name of still store.
    - o LockMode — For future use. Always set to 0.
    - o H/V size — Horizontal and vertical size, Min, max and current value. It shows the current value.
    - o StillState — This tells user if the still is currently being captured or not, or if it is getting deleted.
    - o PngState — The "PNG" for stills are for the thumbnails we capture for the stills.
    - o FileSize — Size of the file created in KBs.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {}, "method":"listStill", "id":"1234", "jsonrpc":"2.0"}

### **takeStill**

- Definition:
  - This API captures/overwrites a still.
- Request:
  - params: {"type": x, "id": y, "file": z}
    - o type — 0 for input source, 1 for BG source.
    - o Id — Index of the source. If the source id of the destination is provided, no still is created and an error is shown.
    - o File — still file id. If you pass "file" : 5, this creates StillStore6.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"type":0, "id": 1, "file": 5}, "method":"takeStill", "id":"1234", "jsonrpc":"2.0"}
  - // This captures a still from input id 1 and saves it as StillStore6.

### **deleteStill**

- Definition:
  - This API deletes a still.
- Request:
  - params: {"id": x}
    - o id—Index of still.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"id": 0}, "method":"deleteStill", "id":"1234", "jsonrpc":"2.0"}

### **listSourceMainBackup**

- Definition:
  - This API shows list of inputs and backgrounds which has backup configured.
- Request:
  - params: {"inputType": x}
    - o x can be - 1(Default): For all Inputs, 0 for inputs and 1 for Background. This is not mandatory parameter, if not provided, all (Inputs + Background) are listed.
- Response:
  - response: Array of inputs / Background  
[{"id":0, "Name":"HDMIInput1", "Backup":[{"id":0, "inputId":8, "stillId":null, "destId":null, "Name":"DPBackground1"}, {"id":2, "inputId":4, "stillId":null, "destId":null, "Name":"DPInput5"}]}
  - o id — Index of inputs or background.
  - o Name — Name of input/Background.
  - o VideoStatus:
    - o 0: there is sync, but cannot acquire / lock mismatch
    - o 1: Video status is OK
    - o 2: Format mismatch between Input config and connector(s)
    - o 3: Capacity / system mode error
    - o 4: A connector lost sync
  - o Array of Backup
    - o Id — backup Index (0, 1, 2). (Max we can set 3 backup)
    - o InputId: index of input which is configured as backup source.
    - o stillId: index of still which is configured as backup source.
    - o destId: index of screen destination which is configured as backup source.
    - o Name: Name of source configured as backup.
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"inputType": -1}, "method":"listSourceMainBackup", "id":"1234", "jsonrpc":"2.0"}

### **activateSourceMainBackup**

- Definition:
  - This API configure backups on inputs and backgrounds.
- Request:
  - params: {"inputId":8,  
          "Backup1": {"SrcType": 1, "SourceId": 1},  
          "Backup2": {"SrcType": 0, "SourceId": 0},  
          "Backup3": {"SrcType": 1, "SourceId": 0},  
          "BackUpState":1}
    - inputId: index of input/Background for which backup needs to be configured.
    - Backup1/Backup2/Backup3:
      - SrcType: 0 for input, 1 for Stills.
      - SourceId: Index of input/background or Still.
    - BackupState: Backup id which needs to be set for backup of the main input. -1 to set primary and is default (If not provided then primary will be activated)
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"inputId":8,  
          "Backup1": {"SrcType": 1, "SourceId": 1},  
          "Backup2": {"SrcType": 0, "SourceId": 0},  
          "Backup3": {"SrcType": 1, "SourceId": 0},  
          "BackUpState":1}, "method":"activateSourceMainBackup", "id":"1234",  
          "jsonrpc":"2.0"}

### **resetSourceMainBackup**

- Definition:
  - This API resets the applied Input backup to primary.
- Request params:
  - params: {"id": 2}
    - id: Input index to be reset.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 22}, "method":"resetSourceMainBackup", "id":"1234",  
          "jsonrpc":"2.0"}

### **listDestinations**

- Definition:
  - This API lists all the destinations with properties such as layers, outputs, id, size, and name.
- Request:
  - params: {"type": x}
    - 0 — Show all the destinations.  
          0 is the default value for the type parameter.
    - 1 — Only screen destinations.
    - 2 — Only aux destinations.
- Response:
  - response: Array of: {"ScreenDestination":[{"id": 0, "Name": "Dest1", "HSize": 3840,  
          "VSize": 1080, "Layers": 1,"DestOutMapColl":[{"id": 0,"DestOutMap":[{"id": 0, "Name":  
          "Out1", "HPos": 0, "VPos": 0, "HSize": 1920, "VSize":1080, "Freeze": 0},{id": 1,  
          "Name": "Out2", "HPos": 1920, "VPos": 0, "HSize": 1920, "VSize":1080, "Freeze":  
          1}]}]}], "AuxDestination":[{"id": 0, "AuxStreamMode": 4}, {"id": 1, "AuxStreamMode":  
          4}]}"
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"type": 0}, "method":"listDestinations", "id":"1234", "jsonrpc":"2.0"}

## **listAuxContent**

- Definition:
  - This API shows Aux destination information.
- Request:
  - params: {"id": x}
    - Id — Index of the Aux destination.
- Response:
  - response: {"id":0,"Name":"AuxDest1","PvwLastSrcIndex":0,"PgmLastSrcIndex":0}
    - id — Index of Aux destination.
    - Name — Name of Aux destination.
    - PvwLastSrcIndex — Input/Background source index in the preview area.
    - PgmLastSrcIndex — Input/Background source index in the program area.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 0}, "method":"listAuxContent", "id":"1234", "jsonrpc":"2.0"}

## **changeAuxContent**

- Definition:
  - This API changes the source in the Aux destinations.
- Request:
  - params: {"id":x, "Name": "AuxDest1", "PvwLastSrcIndex": y, "PgmLastSrcIndex": z}
    - id — Index of the Aux destination.
    - Name — Name of Aux destination. (Optional paramter)
    - PvwLastSrcIndex — Input/Background source index to set in Aux destination in the preview area.
    - PgmLastSrcIndex — Input/Background source index to set in Aux destination in the program area.
    - TestPattern — Provide test pattern id
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params": {"id":0, "Name": "AuxDest1", "PvwLastSrcIndex": 6, "PgmLastSrcIndex": 1}, "method":"changeAuxContent", "id":"1234", "jsonrpc":"2.0"}  
*// Places Source ID 6 into preview and Source ID 1 on program for Aux Destination 1.*
  - {"params":{"id":0, "TestPattern": 3 }, "method":"changeAuxContent", "id":"1234", "jsonrpc":"2.0"}  
*// Turns on the 100% Color Bars Test Pattern in Aux Destination 1.*

## **listContent**

- Definition:
  - This API shows the content of a screen destination.
- Request:
  - params: {"id": x}
    - "id" — Screen destination index.
- Response:
  - response:{"jsonrpc":"2.0","result":{"success":0,"response":{"id":0,"Name":"ScreenDest1","IsActive":1,"BGLyr":[{"id":0,"LastBGSourceIndex":-1, "BGShowMatte":1, "BGColor":{"id":0, "Red":0, "Green":0, "Blue":0}},{id":1, "LastBGSourceIndex":-1, "BGShowMatte":1, "BGColor":{"id":0, "Red":0, "Green":0, "Blue":0}}], "Layers":[{"id":0, "Name":"Layer1-A", "LastSrcIdx":-1, "PvwMode":0, "PgmMode":0, "LinkLayerId":0, "LinkDestId":0, "Capacity":1, "PvwZOrder":0, "PgmZOrder":0, "Freeze":0, "ScalingMode":2, "Window":[{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, {"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}], "Source":[{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, {"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}], "Mask":[{"id":0, "Top":0, "Left":0, "Right":0, "Bottom":0}, {"id":0, "Top":0, "Left":0, "Right":0, "Bottom":0}]}], {"id":1, "Name":"Layer1-B", "LastSrcIdx":-1, "PvwMode":0, "PgmMode":0, "LinkLayerId":1, "LinkDestId":0, "Capacity":1, "PvwZOrder":0, "PgmZOrder":0, "Freeze":0, "ScalingMode":2, "Window":[{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, {"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}], "Source":[{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, {"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}], "Mask":[{"id":0, "Top":0, "Left":0, "Right":0, "Bottom":0}, {"id":0, "Top":0, "Left":0, "Right":0, "Bottom":0}]}], "Transition":[{"id":0, "TransTime":30, "TransPos":0, "ArmMode":1}, {"id":1, "TransTime":30, "TransPos":0, "ArmMode":1}]}}

```
0}}, "OutputCfg": [{"id":0, "Name":"HDMIOutput1", "OutputAOI":[{"id":0, "TestPattern":[{"id":0, "TestPatternMode":0}]}]}], "id":"1234"}
```

- id — index of screen destination.
  - Name — Name of Screen Destination.
  - BGLyr — Background layer index, Last source index of background.  
"id":0 affects the Background in Program. "id":1 affects the Background in Preview.
  - LastBGSourceIndex — This is -1 if no background is dropped, else this is index of last background dropped on screen destination.
  - BGShowMatte — This is if BG to be matte or not.
  - BGColor — This is background color.
  - Layers — Lists layers on screen destination with its properties.
  - Transition — This property of screen destination contains the transition time (from time to move from preview to program).
  - LinkLayerId: Link/Global Layer Index
  - LinkDestId: Link Destination Index
- success: (0=success, anything else is an error)
  - Example:
    - {"params": {"id": 0}, "method":"listContent", "id":"1234", "jsonrpc":"2.0"}

### changeContent

- Definition:
  - This API changes the content of a screen destination by putting background and layers in it.
- Request:
  - params: {"id":0, "TestPattern" :5, "BGLyr":[{"id":0, "LastBGSourceIndex":0, "BGShowMatte":0, "BGColor":[{"id":0, "Red":0, "Green":0, "Blue":0}], {"id":1, "LastBGSourceIndex":0, "BGShowMatte":0, "BGColor":[{"id":0, "Red":0, "Green":0, "Blue":0}]}], "Layers":[{"id":0, "LastSrcIdx":0, "Window":{"HPos":0, "VPos":0, "HSize":400, "VSize":300}, "Source":{"HPos":0, "VPos":0, "HSize":1920, "VSize":1080}, "Mask":{"Left":0.01, "Right":10.1, "Top":0.0, "Bottom":0.0}, "PvwMode":1, "PgmMode":0}]}
  - id — Screen destination index.
  - BGLyr — Background layer index, Last source index of background.  
"id":0 affects the Background in Program. "id":1 affects the Background in Preview.
  - Layers — Layer information collection.
    - id — Layer index number.
    - LastSrcIdx — Source Index number.
    - Window — Layer window size.
    - Source — Source size and offset.
  - Mask — Crop the visible part of the layer. Mask parameters must be sent separately from Window and LastSrcIdx.
  - PvwMode — Set 1 if you want the content in preview. (Default)
  - PgmMode — Set 1 if you want the content in program.
  - TestPattern — Provide test pattern id (0=Off).
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params": {"id":0, "BGLyr":[{"id":0, "LastBGSourceIndex":0, "BGShowMatte":1, "BGColor":[{"id":0, "Red":0, "Green":0, "Blue":0}], {"id":1, "LastBGSourceIndex":0, "BGShowMatte":1, "BGColor":[{"id":0, "Red":0, "Green":1023, "Blue":0}]}], "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}  
*// Sets the Background for Destination 1 to use a black matte on PGM and full green matte on PVW.*
  - {"params": {"id": 0, "Layers": [{"id": 0, "LastSrcIdx":4, "Window":{"HPos": 320, "VPos": 180, "HSize": 1280, "VSize": 720}, "PvwMode":1}], "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}  
*// Places Source ID 4 into Layer 1A on Destination 1's PVW at the given window size and position.*
  - {"params": {"id": 0, "Layers": [{"id": 0, "Mask":{"Left":12.5, "Right":12.5}, "PvwMode":1}], "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}  
*// Sets the Mask values for Layer1A on Destination 1's PVW to 12.5% on the left and right edges.*
  - {"params":{"id":1, "TestPattern":5 }, "method":"changeContent", "id":"1234", "jsonrpc":"2.0"}  
*// Turns on the 32x32 Grid Test Pattern for Destination 2.*

### **listUserKey**

- Definition:
  - This API lists all userkeys in the system.
- Request params:
  - params: {}
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {}, "method": "listUserKeys", "id": "1234", "jsonrpc": "2.0"}

### **recallUserKey**

- Definition:
  - Recall a UserKey on the Event Master processor. User can recall UserKey with id or UserKey name. Send any one of the parameters to recall UserKey.
- Request params:
  - params: {"id": x, "ScreenDestination": [], "Layer": [] }
  - params: {"userkeyName": "UserKeyName", "ScreenDestination": [], "Layer": [] }
    - ScreenDestination—Indexes of screen destinations.
    - Layer—Array of layers index in screen destination on which UserKey is to be recalled.  
*\*Note: If user provides params: {"id": x, "ScreenDestination": [1,2], "Layer": [1,2,3]} then this means Screen 1 – Layer 1,2,3, Screen 2 – Layer 1,2,3 and so on.*
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params": {"id": 0, "ScreenDestination": [0,1,2], "Layer": [0,2,4]}, "method": "recallUserKey", "id": "1234", "jsonrpc": "2.0"}  
*// Applies UserKey id 0 to Layers 1, 3 & 5 on Screens 1, 2 & 3.*
  - {"params": {"userkeyName": "abc", "ScreenDestination": [0,1], "Layer": [0]}, "method": "recallUserKey", "id": "1234", "jsonrpc": "2.0"}  
*// Applies UserKey name "abc" to Layer 1 on Screen 1 & 2.*

### **fillHV**

- Definition:
  - Fits layers to screen destination horizontally and vertically.
- Request:
  - params: {"screenId": x, "Layers": [{"id": 0}, {"id": 1}, {"id": 3}]}
    - screenId — index of screen destination
    - Layers — Array of layer indexes.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"screenId": 0, "Layers": [{"id": 0}, {"id": 1}]}, "method": "fillHV", "id": "1234", "jsonrpc": "2.0"}

### **clearLayers**

- Definition:
  - Clear layers from screen destinations only for custom mode.
- Request:
  - params: {"screenId": x, "Layers": [{"id": 0}, {"id": 1}, {"id": 3}]}
    - screenId — index of screen destination
    - Layers — Array of layer indexes.
- Response:
  - response: null

- success: (0=success, anything else is an error)

- Example:

- {"params": {"screenId": 0, "Layers": [{"id": 0}, {"id": 1}]}, "method": "clearLayers", "id": "1234", "jsonrpc": "2.0"}

### **listSuperAuxContent**

- Definition:

- This API shows the content of a super aux destination.

- Request:

- params: {"id": x}
  - o "id"—Super Aux destination index.

- Response:

- response: {"jsonrpc": "2.0", "result": {"success": 0, "response": {"id": 0, "Name": "SuperAux1", "HDimention": 2, "VDimention": 1, "HSize": 3840, "VSize": 1080, "AuxDestCollection": [{"id": 0, "DestType": 0, "DestXmlId": 0, "Name": "AuxDest1"}, {"id": 1, "DestType": 0, "DestXmlId": 1, "Name": "AuxDest2"}]}}, "id": "1234"}

- o id—index of super aux destination.
- o Name—Name of super aux Destination.
- o HDimention — Horizontal dimension of super aux destination
- o VDimention — Vertical dimension of super aux destination
- o HSize: Horizontal size of super aux destination
- o VSize: Vertical size of super aux destination
- o AuxDestCollection —Array of aux destination with index and name of aux destination and destination type.

- success: (0=success, anything else is an error)

- Example:

- {"params": {"id": 0}, "method": "listSuperAuxContent", "id": "1234", "jsonrpc": "2.0"}

### **changeSuperAuxContent**

- Definition:

- This API changes sources for any aux destination which is part of super aux.

- Request:

- Params: {"id": 0, "Destinations": [{"id": 0, "Name": "AuxDest1", "PvwLastSrcIndex": 0, "PgmLastSrcIndex": 0}]}
  - o id—Super Aux destination index.
  - o Destinations — Array of Aux destination with index, Name of aux destination to be renamed, Preview source and Program source to be modified in aux destination.
  - o TestPattern — Test Pattern index of Aux Destination to be changed (0=Off).

- Response:

- response: null
- success: (0=success, anything else is an error)

- Examples:

- {"params": {"id": 0, "Destinations": [{"id": 0, "Name": "AuxDest1", "PvwLastSrcIndex": 0, "PgmLastSrcIndex": 0}]}, "method": "changeSuperAuxContent", "id": "1234", "jsonrpc": "2.0"}  
// Places source id 0 on both preview and program of Aux Destination 1.
- {"params": {"id": 0, "Destinations": [{"id": 0, "Name": "AuxDest1", "PvwLastSrcIndex": 10}, {"id": 1, "Name": "AuxDest2", "PvwLastSrcIndex": 11}]}, "method": "changeSuperAuxContent", "id": "1234", "jsonrpc": "2.0"}  
// Places source id 10 in preview of Aux Destination 1 and source id 11 in preview of Aux Destination 2 of the SuperAux .
- {"params": {"id": 0, "Destinations": [{"id": 0, "Name": "AuxDest1", "TestPattern": 5}]}, "method": "changeSuperAuxContent", "id": "1234", "jsonrpc": "2.0"}  
// Turns on the 32x32 Grid Test Pattern for Aux Destination 1 in the SuperAux.

### **listSuperDestContent**

- Definition:

- This API shows the content of a super screen destination.

- Request:

P 13 / 22

- params: {"id": x}
  - o "id"—Super/Link Screen destination index.
- Response:
  - response:{"jsonrpc":"2.0","result":{"success":0,"response":{"id":0,"Name":"SuperDest1","HDimention":1,"VDimention":1,"HSize":1920,"VSize":1080,"GlobalLayers":2,"DestCollection":[{"id":0,"DestType":1,"DestXmlId":0,"Name":"ScreenDest1"}],"GlobalLayerCollection":{"GlobalLayer":[{"id":0,"Name":"SuperLayer1-A","SuperLayerLinkedState":1,"LinkLayer":[{"LinkLayerXmlId":0,"DestXmlId":0}]},"id":1,"Name":"SuperLayer1-B","SuperLayerLinkedState":1,"LinkLayer":[{"LinkLayerXmlId":1,"DestXmlId":0}]}]}}, {"id":"1234"}
    - o id — index of super screen destination.
    - o Name — Name of super Screen Destination.
    - o HDimention — Horizontal dimension of super screen destination
    - o VDimention — Vertical dimension of super screen destination
    - o HSize: Horizontal size of super screen destination
    - o VSize: Vertical size of super screen destination
    - o GlobalLayers — Count of global Layer.
    - o DestCollection — Array of screen destination with index and name of screen destination and destination type.
    - o GlobalLayerCollection — Array of global layer with information of index, name of link layer.
    - o SuperLayerLinkedState — State of super/Link Layer.
    - o LinkLayerXmlId: Link Layer Index
    - o DestXmlId: Link layer part of screen Destination Index.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 0}, "method":"listSuperDestContent", "id":"1234", "jsonrpc":"2.0"}

### **changeSuperDestContent**

- Definition:
  - This API changes layer parameters for each global super layer in all screen destinations that are part of a super destination.
- Request:
  - Params: {"id":0,"GlobalLayers":
    - [{"id":0,"Window":{"HPos":0,"VPos":0,"HSize":700,"VSize":300}}]}
    - o id — Super Screen destination index.
    - o GlobalLayers — Array of Global Layers with index,
      - o Window — H/V position and H/V size.
      - o LastSrcIdx — Source index to be used in the Global Layer
      - o Mask — Mask position of the Global Layer with left, right, top, and bottom percentage. Mask parameters must be sent separately from Window and LastSrcIdx.
      - o PvwMode — Set 1 if you want the change on Preview (Default).
      - o PgmMode — Set 1 if you want the change on Program.
      - o TestPattern — Test Pattern index for the SuperDestination to be changed.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params":{"id":0,"GlobalLayers":[{"id":0,"LastSrcIdx":10, "Window":{"HPos":100,"VPos":200,"HSize":700,"VSize":300},"PvwMode":0,"PgmMode":1},{"id":2,"LastSrcIdx":12, "Window":{"HPos":1600,"VPos":200,"HSize":1280,"VSize":720},"PvwMode":0,"PgmMode":1}]},"method":"changeSuperDestContent", "id":"1234", "jsonrpc":"2.0"}
    - // For SuperDest1, places source ID 10 into SuperLayer1a and source ID 12 into SuperLayer2a on program.*
  - {"params":{"id":0,"GlobalLayers":[{"id":0, "Mask":{"Left":10,"Right":10,"Top":10,"Bottom":5}]},"method":"changeSuperDestContent", "id":"1234", "jsonrpc":"2.0"}
    - // Sets the masking percentage to the declared values for SuperLayer1 in SuperDest1.*
  - {"params":{"id":0,"TestPattern":5}, "method":"changeSuperDestContent", "id":"1234", "jsonrpc":"2.0"}
    - // Turns on the 32x32 Grid Test Pattern for All Destinations in SuperDest1.*

### ***listPresets***

- Definition:
  - This queries the list of Presets on a particular destination or on the system.
- Request:
  - `params: {"ScreenDest":x, "AuxDest":x},`  
"x" can be:
    - -2: Do not include any destinations of this type. (Has priority over particular id, if passed as a parameter.)
    - -1: Do not care (All presets). (Has priority over particular id, if passed as a parameter.)
    - 0-999: want to see the presets with the destination's particular id in it or array of ids. E.g. `"ScreenDestination":[{"id": 2}, {"id": 3}]`
- Response:
  - `response: Array of: [{"id": 0, "Name": "Preset3.00", "LockMode": 0, "presetSno": 3.00}, {"id": 1, "Name": "Preset4.00", "LockMode": 0, "presetSno": 4.00}]`
    - Response contains the array of presets. Above response contains id, name, lock mode preset serial number of the all the presets.
  - `success: (0=success, anything else is an error)`
- Example:
  - `{"params":{"ScreenDest": 0}, "method":"listPresets", "id":"1234", "jsonrpc":"2.0"}`

### ***savePreset***

- Definition:
  - Creates a Preset on the Event Master processor.
- Request:
  - `params: {"presetName": "NewPreset", "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}`
  - `params: {"presetName": "NewPreset", "serialNo": 1.01, "saveFromProgram":1, "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}`
    - `presetName` — Name of the Preset to save.
    - `ScreenDestinations` — `ScreenDest` id for the Preset to be created.
    - `AuxDestinations` — `AuxDest` id for the Preset to be created.
    - `ScreenDestination`, `AuxDestinations` are optional parameters. If user doesn't provide it, Preset will be saved for selected destinations.
    - `serialNo` — serial number for the preset to be saved. If preset exist, it will be overwritten. (Optional). Only 2-Digit decimal points are recommended, If user provides more than 2 decimal places then the number may be round off to 2-digit decimal place (hundredths).
    - `saveFromProgram` — This flag is set to 1 if the preset is to be saved from program, else the default will be saved from preview. (Optional)
- Multi-Operator Mode:

New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.

  - `params: {"presetName": "NewPreset", "operatorId": y} (for normal operator)`
    - "operatorId" — operator index (For current release only 0, 1, 2 are valid indexes).
    - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
  - `params: {"presetName": "NewPreset", "password": "xyz"} (for super operator)`
    - `password` — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
- Response:
  - `response: null`
  - `success: (0=success, anything else is an error)`
- Examples:
  - `{"params": {"presetName": "NewPreset"}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetName": "NewPreset", "ScreenDestination": {"id": 0}, "AuxDestination":{"id": 0}}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`
  - `{"params": {"presetName": "NewPreset", "serialNo": 1.01, "saveFromProgram":1}, "method":"savePreset", "id":"1234", "jsonrpc":"2.0"}`

For normal operator:

- {"params": {"presetName": "NewPreset", "serialNo": 5.00, "operatorId": 2}, "method": "savePreset", "id": "1234", "jsonrpc": "2.0"}

For super operator:

- {"params": {"presetName": "NewPreset", "operatorId": 2}, "method": "savePreset", "id": "1234", "jsonrpc": "2.0"}

### **renamePreset**

- Definition:
  - Rename a Preset on the Event Master processor. User can rename Preset with id, Preset serial number, or Preset name.  
Send any one of the parameters to rename Preset.
- Request params:
  - params: {"id": x, "newPresetName": "NewPresetName"}
  - params: {"presetSno": x.y, "newPresetName": "NewPresetName"}
  - params: {"presetName": "OldPresetName", "newPresetName": "NewPresetName"}
    - o "newPresetName" — New Preset name to set.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params": {"id": 0, "newPresetName": " newPresetName "}, "method": "renamePreset", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"presetName": "NewPreset", "newPresetName": "NewPresetName"}, "method": "renamePreset", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"presetSno": 1.00, "newPresetName": " newPresetName "}, "method": "renamePreset", "id": "1234", "jsonrpc": "2.0"}

### **deletePreset**

- Definition:
    - Delete a Preset on the Event Master processor.  
User can delete Preset with id, Preset serial number, or Preset name.  
Send any one of the parameters to delete Preset.
  - Request:
    - params: {"id": x}
    - params: {"presetSno": x.y}
    - params: {"presetName": "PresetName"}
  - Multi-Operator Mode:

New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.

    - params: {"id": x, "operatorId": y} (for normal operator)
      - o "operatorId" — operator index (For current release only 0, 1, 2 are valid indexes).
      - o If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
    - params: {"id": x, "password": "xyz"} (for super operator)
      - o password — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
  - Response:
    - response: null
    - success: (0=success, anything else is an error)
  - Examples:
    - {"params": {"id": 1}, "method": "deletePreset", "id": "1234", "jsonrpc": "2.0"}
    - {"params": {"presetSno": 1.00}, "method": "deletePreset", "id": "1234", "jsonrpc": "2.0"}
    - {"params": {"presetName": "Preset 5.00"}, "method": "deletePreset", "id": "1234", "jsonrpc": "2.0"}
- For super operator:
- {"params": {"id": 6, "password": "123"}, "method": "deletePreset", "id": "1234", "jsonrpc": "2.0"}

For normal operator:

- {"params": {"id": 5, "operatorId": 2}, "method": "deletePreset", "id": "1234", "jsonrpc": "2.0"}

### **activatePreset**

- Definition:
    - Recall a Preset on the Event Master processor. User can recall Preset with id, Preset serial number, or Preset name.  
Send any one of the parameters to recall Preset.
  - Request params:
    - params: {"id": x, "type": x}
    - params: {"presetSno": x.y, "type": x}
    - params: {"presetName": "PresetName"}
      - o "type"—0 to recall in preview (default), 1 to recall in program. This is not a mandatory parameter but should be given when the user wants to recall a Preset in program.
  - Multi-Operator Mode:
    - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.
    - params: {"id": x, "operatorId": y} (for normal operator)
      - o "operatorId" — operator index (For current release only 0, 1, 2 are valid indexes).
      - o If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
    - params: {"id": x, "password": "xyz"} (for super operator)
      - o password — Super user password saved. When this is passed, actions will be performed as if no operator is enabled.
  - Response:
    - response: null
    - success: (0=success, anything else is an error)
  - Examples:
    - {"params": {"id": 0, "type": 0}, "method": "activatePreset", "id": "1234", "jsonrpc": "2.0"} //Recall in preview with id 0.
    - {"params": {"presetName": "abc"}, "method": "activatePreset", "id": "1234", "jsonrpc": "2.0"} //Recall in preview with preset name "abc".
    - {"params": {"presetSno": 1.00, "type": 1}, "method": "activatePreset", "id": "1234", "jsonrpc": "2.0"} //Recall in program with presetSno 1.
- For super operator:
- {"params": {"id": 6, "password": "123"}, "method": "activatePreset", "id": "1234", "jsonrpc": "2.0"}
- For normal operator:
- {"params": {"id": 5, "operatorId": 2}, "method": "activatePreset", "id": "1234", "jsonrpc": "2.0"}

### **recallNextPreset**

- Definition:
  - Recall the next Preset on the Event Master processor.
  - No parameter is required.
  - Make sure that the user has at least recalled one Preset. Web app recalls the next Preset following the last Preset recalled.
- Request:
  - params: {}
- Response:
  - response: null
  - success: (0=success, anything else is an error)  
An error is shown if there was no previously recalled Preset or if there is no next Preset in the list.
- Example:
  - {"params": {}, "method": "recallNextPreset", "id": "1234", "jsonrpc": "2.0"}

### ***listDestinationsForPreset***

- Definition:
  - Lists the content of a Preset.
- Request:
  - params: {"id":x},  
"x" can be:
    - -1: List all Presets.
    - 0-999: list only that specific Preset.
- Response:
  - response: Array of:[{"id": 0, "Name": "Preset3.00", "LockMode": 0, "presetSno": 3.00, "ScreenDest":[{"id": 0}, {"id": 3}], "AuxDest":[{"id": 0}, {"id": 1}]]
    - Response contains the array of Presets.
  - success: (0=success, anything else is an error)
- Example:
  - {"params":{"id": 0}, "method":"listDestinationsForPreset", "id":"1234", "jsonrpc":"2.0"}

### ***freezeDestSource***

- Definition:
  - This API Freezes/Unfreezes the sources.
- Request:
  - params: {"type": x, "id": y, "screengroup": z, "mode": 0/1}
    - type — type of source.
      - 0 — Input source.
      - 1 — Background source.
      - 2 — ScreenDestination.
      - 3 — AuxDestination.
    - id — Index of the source.
    - Screengroup — For future use. Always set to 0.
    - Mode — 0: UnFreeze, 1: Freeze.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"type":0, "id":0, "screengroup":0, "mode":1}, "method":"freezeDestSource", "id":"1234", "jsonrpc":"2.0"}

### ***armUnarmDestination***

- Definition:
  - Arm and Unarm the destinations for transitions and saving presets.
- Request:
  - params: {"arm": 1, "ScreenDestination":[{"id": 2}, {"id": 3}], "AuxDestination":[{"id": 1}, {"id": 2}]}
    - "arm": "x" can be: 0 to unarm and 1 to arm.
    - ScreenDestinations — ScreenDest ids to arm/unarm.
    - AuxDestinations — AuxDest ids to arm/unarm.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"arm": 1, "ScreenDestination":[{"id": 0}, {"id": 2}], "AuxDestination":[{"id": 0}, {"id": 1}], "method":"armUnarmDestination", "id":"1234", "jsonrpc":"2.0"}
  - {"params": {"arm": 0, "ScreenDestination":[{"id": 0}, {"id": 2}], "AuxDestination":[{"id": 0}, {"id": 1}], "method":"armUnarmDestination", "id":"1234", "jsonrpc":"2.0"}

## **activateDestGroup**

- Definition
  - Recall a DestGroup on the Event Master processor. User can recall DestGroup with id, DestGroup serial number, or DestGroup name.
  - Send any one of the parameters to recall DestGroup.
- Request params:
  - params: {"id": x}
  - params: {"destGrpSno": x.y}
  - params: {"destGrName": "GroupName"}
    - o id – Index of the Destination group.
    - o destGrpSno – Destination group serial number
    - o destGrName – Destination group name.
- Response:
  - Response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 0}, "method": "activateDestGroup", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"destGrpName": "abc"}, "method": "activateDestGroup", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"destGrpSno": 1.00}, "method": "activateDestGroup", "id": "1234", "jsonrpc": "2.0"}

## **allTrans**

- Definition
  - It executes the "allTrans" command.
  - If multi-operator mode is enabled, all-trans will affect only on those destinations which are selected for the requested operator.
- Request:
  - param: {"transTime": 40} - integer value, will be applied to all armed destinations. (optional)
- Multi-Operator Mode:
  - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are required only when one or more operators are enabled.
  - params: {"operatorId": y} (for normal operator)
    - o "operatorId"— operator index (For current release only 0, 1, 2 are valid indexes).
    - o If user still want to use "super-operator" mode, password is required which is passed as a parameter.
  - params: {"password": "xyz"} (for super operator)
    - o password — Super user password saved. When this is passed, actions will be performed as if no other operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Examples:
  - {"params": {}, "method": "allTrans", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"transTime": 40}, "method": "allTrans", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"operatorId": 1}, "method": "allTrans", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"password": "123"}, "method": "allTrans", "id": "1234", "jsonrpc": "2.0"}

## **cut**

- Definition
  - It executes the "Cut" command.
  - If multi-operator mode is enabled, cut will affect only on those destinations which are selected for the requested operator.
- Request:
  - params: {} - It doesn't require any parameter.
- Multi-Operator Mode:
  - New parameters are introduced to work with multi-operator mode along with the above parameters. These parameters are used only when one or more operators are enabled.
  - params: {"operatorId": y} (for normal operator)

- "operatorId" — operator index (For current release only 0, 1, 2 are valid indexes).
  - If user wants to use "super-operator" mode, its password is required which is passed as a parameter.
- params: {"password": "xyz"} (for super operator)
  - password — Super user password saved. When this is passed, actions will be performed as if no other operator is enabled.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example
  - {"params": {}, "method": "cut", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"operatorId": 1}, "method": "cut", "id": "1234", "jsonrpc": "2.0"}
  - {"params": {"password": "123"}, "method": "cut", "id": "1234", "jsonrpc": "2.0"}

### ***listCues***

- Definition:
  - List all the cues.
- Request:
  - params: {"id": 1}, "method": "listCues", "id": "1234", "jsonrpc": "2.0"}
    - id — Index of the input config.
- Response:
  - Response: Array of cue objects.
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {}, "method": "listCues", "id": "1234", "jsonrpc": "2.0"}

### ***activateCue***

- Definition:
  - This API gives option to play/pause/stop cue.
- Request:
  - params: {"id": 1, "type": x}, "method": "activateCue", "id": "1234", "jsonrpc": "2.0"}
    - id — Index of the cue.
    - type — (Default is play). x" can be: 0 — Play. 0 is the default value for the type parameter. 1 — Pause, 2 — Stop
- Response:
  - Response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"id": 1}, "method": "activateCue", "id": "1234", "jsonrpc": "2.0"}  
//Play – no params or type 0
  - {"params": {"type": 1}, "method": "activateCue", "id": "1234", "jsonrpc": "2.0"}  
//Pause – type 1
  - {"params": {"type": 2}, "method": "activateCue", "id": "1234", "jsonrpc": "2.0"}  
//Stop – type 2

### ***mvrLayoutChange***

- Definition:
  - This API changes layout in the given frame multiviewer.
- Request:
  - params: {"frameUnitId": x, "mvrLayoutId": x}
    - frameUnitId: Frame unit id for which MVR layout needs to be changed.
    - mvrLayoutId: Mvr layout index. Possible value 0 to 3.
    - Both are mandatory parameters for this API.
- Response:
  - response: null
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"frameUnitId": 0, "mvrLayoutId": 1}, "method": "mvrLayoutChange", "id": "1234", "jsonrpc": "2.0"}

## Subscription and Un-Subscription

When a subscription is done from a JSON-based application, a notification is sent to the ip port where the application is running when there is change for which the user has subscribed.

Actual notification is sent asynchronously as an HTTP Post, with the following structure: {result: {method: "notification", notificationType: "ScreenDestChanged", change: { add: [2], remove: [], update: [0, 1, 2] }}}.

The change field contains the XmlId(s) of the screens that were added/removed or updated.



**All subscriptions are lost once the Event Master processor is restarted, and they must be subscribed again if required.**

### ***subscribe***

- Definition
  - User can use this API to subscribe to change events in the Event Master processor.
  - Once subscribed, the API sends a notification in the form of an HTTP Post to the Url: http://hostname:port/.
- Request:
  - params: {"hostname": hostname, "port": port, "notification": notificationType[]}
    - hostname — Hostname or IP Address to which the notifications are sent.
    - port — TCP port to which the notification are posted.
    - notificationTypes — an array of notifications to which a user wants to subscribe.
      - ScreenDestChanged
      - AUXDestChanged
      - FrameChanged
      - NativeRateChanged
      - InputCfgChanged
      - SourceChanged
      - BGSourceChanged
      - PresetChanged
      - StillChanged
      - OutputCfgChanged
      - CueChanged
- Response:
  - response: {"method": "subscribe"}
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"hostname": "192.168.247.131", "port": "3000", "notification": ["ScreenDestChanged", "AUXDestChanged"]}, "method": "subscribe", "id": "1234", "jsonrpc": "2.0"}

### ***unsubscribe***

- Definition
  - User can use this API to remove the subscription for the given hostname: port and notificationType.
- Request:
  - params: {"hostname":hostname, "port":port, "notification":notificationType[]}
    - hostname — Hostname or IP Address from which the subscription is to be removed.
    - port — TCP port.
    - notificationTypes — an array of notifications to which a user wants to subscribe.
      - ScreenDestChanged
      - AUXDestChanged
      - FrameChanged
      - NativeRateChanged
      - InputCfgChanged
      - SourceChanged
      - BGSourceChanged
      - PresetChanged
      - StillChanged
      - OutputCfgChanged
      - CueChanged

- Response:
  - response: {"method": "unsubscribe"}
  - success: (0=success, anything else is an error)
- Example:
  - {"params": {"hostname": "192.168.247.131", "port": "3000", "notification": ["ScreenDestChanged", "AUXDestChanged"]}, "method": "unsubscribe", "id": "1234", "jsonrpc": "2.0"}