Monitor Wall JSON-RPC Reference

Version 11.10

Contents

General Notes 3
Glossary 4
JSON-RPC Server 5
Example 5
Data Sets in JSON Schema 6
Screen Rectangle Data 6
Layout Data 6
Digital Zoom Data 7
Dewarping Zoom Data 7
PTZ Speed Data 7
Input Data 8
Output Data 8
Source Data 8
Connection Data 9
Banner Salvo Data 9
Requests in JSON Schema 10
System Get Info 10
System Get Date and Time 10
Connections Get 11
Connection Open 12
Connection Close 12
Connections Close All 12
Output Set Highlighting 12
Output Clear Highlighting 12
Ouput Get Zoom 13
Output Store PTZ Preset 13
Output Set PTZ Preset 13

Output Set PTZ Speeds 14
Output Freeze 14
Output Un-Freeze 14
Line Enter Fullscreen 14
Line Leave Fullscreen 15
Line Set Layout ID 15
Line Get Layout ID 15
Line Get Layout 16
Line Get Coders 16
Lines Get 17
Lines Freeze All 17
Lines Un-Freeze 17
Lines Store Favorite 17
Lines Set Favorite 17
Banners Get Banner Salvo IDs 18
Banners Get Banner Salvos 18
Banners Set Banner Salvos 18
Banners Stop All Banner Salvos 19
Inputs Get 19
Outputs Get 19
Salvos Get 20

General Notes

Copyright
This manual is the intellectual property of IQSIGHT and is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted for any purpose, by whatever means, be they electronic or mechanical, without the express written permission of IQSIGHT.
© Copyright 2026 IQSIGHT

Note
This manual was compiled with the greatest of care and all information double checked. At the time of printing the description was complete and correct. Because of the further development of products, the content of the manual might change without prior notice. IQSIGHT will not be liable for damage which is directly or indirectly due to errors, incompleteness, or discrepancies between the manual and the product described.

Trade marks
All names used in this manual for hardw are and software are very probably registered trade marks and must be treated as such.

Glossary

Hardware Decoder	Physical device that hosts one instance of the Monitor Wall application.
Monitor Wall	The name of the main application designed for showing streams (video, audio and metadata) from different video sources.
Line	One-based index for physical video out connectors of a hardware decoder. Each line corresponds thus to one connected physical screen.
Coder	One-based index for video panels on each line. Each video panel can display one video stream.
Layout	Arrangement of video panels (coders) on a line. Each line can have its own layout.
IP Matrix	Cluster of 1 to 4 hardware decoders that control up to 8 physical screens.

JSON-RPC Server

The Monitor Wall's JSON-RPC server implements the JSON-RPC specification 2.0. JSON remote procedure calls can be executed by posting JSON requests with the according arguments to the Monitor Wall's http or https server using basic user name and password authentication.

The scope of the JSON requests is restricted to live operations which are supported with "user" credentials, configuration operations that require "service" credentials are only possible via the IQSIGHT proprietary remote control protocol (RCP+), described in a separate document.

Example

Posting the System.GetInfo request to the Monitor Wall server's endpoint "<myUrl>/api/monitorwall/v1"

```
{
  "id": 1234,
  "jsonrpc": "2.0",
  "method": "System.GetInfo"
}
```

will trigger the response

```
{
  "jsonrpc": "2.0",
  "result": {
    "unitName": "...",
    "unitId": "...",
    "serialNumber": "...",
    "softwareVersion": "10.00.0000",
    "productName": "Monitor Wall",
    "hardwareVersion": "F100FE40",
    "variantId": 0,
    "installationCode": "...",
    "licenseInfo": "MonitorWall license [authorization number ...]"
  },
  "id": 1234
}
```

As an example, the http POST can be executed via Microsoft's Powershell with the following code snippet:

```
$user = "user"
$password = "<myPassword>"
$pair = "${user}:${password}"
$bytes = [System.Text.Encoding]::ASCII.GetBytes($pair)
$base64 = [System.Convert]::ToBase64String($bytes)
$basicAuthentication = "Basic $base64"
$headers = @{ Authorization = $basicAuthentication }
$endpoint = "http://<myIPv4>:80/api/monitorwall/v1"
$request = '{"id":0,"jsonrpc":"2.0","method":"System.GetInfo"}'
```

```
$response = Invoke-WebRequest -uri "$endpoint" -Headers $headers -Method POST -ContentType
"application/json" -Body "$request"
```

In order to post the request via a secure connection, the code line that defines the endpoint needs to be replaced with the following two lines:

```
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
$endpoint = "https://<myIPv4>:443/api/monitorwall/v1"
```

The server certificate validation callback implementation allows connections to servers with self-signed certificates.

Further examples can be found in the ApiTestTool C# example.

Data Sets in JSON Schema

The data sets in this chapter are referenced by the Monitor Wall's JSON-RPC requests.

Screen Rectangle Data

Bounding box of a rectangular screen area in pixel.

```
"ScreenRectangleData":
{
        "required": [ "leftPx", "topPx", "widthPx", "heightPx" ],
        "properties":
        {
                "leftPx": { "type": "integer", "minimum": -16384, "maximum": 16384 },
                "topPx": { "type": "integer", "minimum": -16384, "maximum": 16384 },
                "widthPx": { "type": "integer", "minimum": 0, "maximum": 16384 },
                "heightPx": { "type": "integer", "minimum": 0, "maximum": 16384 }
        }
}
```

Layout Data

Tile layout descriptor for a specific line.

```
"LayoutData":
{
        "required":
        [
                "lineId", "layoutId", "columnCount", "rowCount",
                "thumbnailColumnCount", "thumbnailRowCount", "thumbnailFullColumnCount",
                "thumbnailFullRowCount", "screenAspectX", "screenAspectY",
                "tileAspectX", "tileAspectY"
        ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 0, "maximum": 9 },
```

```
                "layoutId": { "type": "integer", "minimum": 0, "maximum": 2147483647 },
                "columnCount": { "type": "integer", "minimum": 1, "maximum": 9 },
                "rowCount": { "type": "integer", "minimum": 1, "maximum": 9 },
                "thumbnailColumnCount": { "type": "integer", "minimum": 0, "maximum": 99 },
                "thumbnailRowCount": { "type": "integer", "minimum": 0, "maximum": 99 },
                "thumbnailFullColumnCount": { "type": "integer", "minimum": 0, "maximum": 99 },
                "thumbnailFullRowCount": { "type": "integer", "minimum": 0, "maximum": 99 },
                "screenAspectX": { "type": "integer", "minimum": 1, "maximum": 16384 },
                "screenAspectY": { "type": "integer", "minimum": 1, "maximum": 16384 },
                "tileAspectX": { "type": "integer", "minimum": 1, "maximum": 16384 },
                "tileAspectY": { "type": "integer", "minimum": 1, "maximum": 16384 }
        }
}
```

Digital Zoom Data

Zoom settings for a tile showing a standard video stream.

```
"DigitalZoomData":
{
        "type": "object",
        "required": [ "centerX", "centerY", "scale" ],
        "properties":
        {
                "centerX": { "type": "number", "minimum": 0.0, "maximum": 1.0 },
                "centerY": { "type": "number", "minimum": 0.0, "maximum": 1.0 },
                "scale": { "type": "number", "minimum": 1.0, "maximum": 32.0 }
        }
}
```

Dewarping Zoom Data

View direction and zoom settings for a tile showing a panoramic video stream. Angles are defined in radians.

```
"DewarpingZoomData":
{
        "type": "object",
        "required": [ "pan", "tilt", "scale" ],
        "properties":
        {
                "pan": { "type": "number", "minimum": -6.284, "maximum": 6.284 },
                "tilt": { "type": "number", "minimum": -6.284, "maximum": 6.284 },
                "scale": { "type": "number", "minimum": 0.0, "maximum": 32.0 }
        }
}
```

PTZ Speed Data

Relative speed settings for pan, tilt, zoom, iris, and focus.

```
"PTZSpeedsData":
{
```

```
        "type": "object",
        "properties":
        {
                "pan": { "type": "number", "minimum": -1.0, "maximum": 1.0 },
                "tilt": { "type": "number", "minimum": -1.0, "maximum": 1.0 },
                "zoom": { "type": "number", "minimum": -1.0, "maximum": 1.0 },
                "iris": { "type": "number", "minimum": -1.0, "maximum": 1.0 },
                "focus": { "type": "number", "minimum": -1.0, "maximum": 1.0 }
        }
}
```

Input Data

Video source descriptor.

```
"InputData":
{
        "required": [ "url", "type" ],
        "properties":
        {
                "url": { "type": "string", "format": "uri" },
                "type": { "type": "string", "enum": [ "VIP", "ONVIF", "RTSP" ] },
                "lineId": { "type": "integer", "minimum": 0, "maximum": 9 },
                "coderId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                "protocol": { "type": "string", "enum": [ "UDP", "TCP" ] },
                "forceUnicast": { "type": "boolean" },
                "audio": { "type": "integer", "minimum": 0, "maximum": 4095 },
                "ptzPresetId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                "recording": { "type": "integer", "minimum": 0, "maximum": 4095 }
        }
}
```

Output Data

Video panel (tile, coder) selector.

```
"OutputData":
{
        "required": [ "lineId", "coderId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                "coderId": { "type": "integer", "minimum": 1, "maximum": 64 }
        }
}
```

Source Data

Video source selector.

```
"SourceData":
{
```

```
        "oneOf":
        [
                {
                        "required": ["inputId"],
                        "properties":
                        {
                                "inputId": { "type": "integer", "minimum": 0, "maximum": 4095 }
                        }
                },
                {
                        "required": ["salvoId"],
                        "properties":
                        {
                                "salvoId": { "type": "integer", "minimum": 0, "maximum": 4095 }
                        }
                },
                {
                        "required": ["input"],
                        "properties":
                        {
                                "input": { "$ref": "#/definitions/InputData" }
                        }
                }
        ]
}
```

Connection Data

Video connection descriptor.

```
"ConnectionData":
{
        "required": [ "source", "output" ],
        "properties":
        {
                "source": { "$ref": "#/definitions/SourceData" },
                "output": { "$ref": "#/definitions/OutputData" },
                "digitalZoom": { "$ref": "#/definitions/DigitalZoomData" },
                "dewarpingZoom": { "$ref": "#/definitions/DewarpingZoomData" }
        }
}
```

Banner Salvo Data

Banner salvo descriptor.

```
"BannerSalvoData":
{
        "required": [ "lineId", "position", "bannerSalvoId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
```

```
                "position": { "type": "string", "enum": [ "Top", "Center", "Bottom" ], "default":
"Center" },
                "bannerSalvoId": { "type": "integer", "minimum": 0, "maximum": 16 }
        }
}
```

Requests in JSON Schema

The requests in this chapter are referenced by the Monitor Wall's JSON-RPC requests.

System Get Info

Retrieves basic information about Monitor Wall server. See initial example.

```
"System.GetInfo.Request": {}

"System.GetInfo.Response":
{
        "required":
        [
                "unitName", "unitId", "serialNumber", "softwareVersion", "productName",
                "hardwareVersion", "variantId", "installationCode", "licenseInfo"
        ],
        "properties":
        {
                "unitName": { "type": "string" },
                "unitId": { "type": "string" },
                "serialNumber": { "type": "string" },
                "softwareVersion": { "type": "string" },
                "productName": { "type": "string" },
                "hardwareVersion": { "type": "string" },
                "variantId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                "installationCode": { "type": "string" },
                "licenseInfo": { "type": "string" },
                "hostname": { "type": "string" }
        }
}
```

System Get Date and Time

Retrieves information about the current date and time. The date is formatted as YYYY-MM-DD, the time is formatted as HH:mm:ss, the local time offset is given in seconds. The timezone name is only supported in English. The NTP server is only included when configured.

```
"System.GetDateTime.Request": {}

"System.GetDateTime.Response":
{
        "required": [ "date", "timeUtc", "localTimeOffsetS", "timezone" ],
        "properties":
        {
                "date": { "type": "string" },
                "timeUtc": { "type": "string" },
                "localTimeOffsetS": { "type": "integer", "minimum": -86400, "maximum": 86400 },
                "timezone": { "type": "string" },
                "ntpServer": { "type": "string" }
        }
}
```

Connections Get

Retrieves descriptors for all active video streaming connections

```
"Connections.Get.Request": {}

"Connections.Get.Response":
{
	"type": "array",
	"items": { "$ref": "#/definitions/ConnectionData" }
}
```

Here is an example on how the schema translates into an actual JSON-RPC request and response for a Monitor Wall server with two cameras A and B streaming to the first two video panels of line 1:

Request:

```
{
	"id": 2345,
	"jsonrpc": "2.0",
	"method": "Connections.Get"
}
```

Response:

```
{
	"jsonrpc": "2.0",
	"result":
	[
		{
			"source":
			{
				"input":
				{
					"url": "rcpp://user:*@<myIPv4_camA>",
					"type": "VIP",
					"lineId": 1,
					"coderId": 0,
					"audio": 1
				}
			},
			"output":
			{
				"lineId": 1,
				"coderId": 1
			}
		},
		{
			"source":
			{
				"input":
				{
					"url": "rcpp://user:*@<myIPv4_camB>",
					"type": "VIP",
					"lineId": 1,
					"coderId": 0,
					"audio": 1
				}
			},
			"output":
			{
				"lineId": 1,
				"coderId": 2
			}
		}
```

```
        ],
        "id": 2345
}
```

Please note that input line and coder IDs refer to camera lines and coders (encoders), and output line and coder IDs refer to Monitor Wall lines (screens) and coders (video panels, tiles, decoders).

Connection Open

Connects a selected video source to a selected video panel.

```
"Connection.Open.Request":
{
        "$ref": "#/definitions/ConnectionData"
}

"Connection.Open.Response": {}
```

Connection Close

Closes the active connection of a selected video panel.

```
"Connection.Close.Request":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" }
        }
}

"Connection.Close.Response": {}
```

Connections Close All

Closes all active connections.

```
"Connections.CloseAll.Request": {}

"Connections.CloseAll.Response": {}
```

Output Set Highlighting

Activates one of four independent highlighting rectangles for a selected video panel.

```
"Output.SetHighlighting.Request":
{
        "required": [ "output", "highlightingIndex" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "highlightingIndex": { "type": "integer", "minimum": 1, "maximum": 4 }
        }
}

"Output.SetHighlighting.Response": {}
```

Output Clear Highlighting

Deactivates either all or one of four highlighting rectangles. All will be cleared if the optional highlighting index is omitted.

```
"Output.ClearHighlighting.Request":
```

```
{
        "properties":
        {
                "highlightingIndex": { "type": "integer", "minimum": 1, "maximum": 4 }
        }
}

"Output.ClearHighlighting.Response": {}
```

Ouput Get Zoom

Retrieves the current zoom settings for a selected video panel. This can be either a digital zoom setting for standard cameras or a dewarping zoom settings for panoramic cameras.

```
"Output.GetZoom.Request":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" }
        }
}

"Output.GetZoom.Response":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "digitalZoom": { "$ref": "#/definitions/DigitalZoomData" },
                "dewarpingZoom": { "$ref": "#/definitions/DewarpingZoomData" }
        }
}
```

Output Store PTZ Preset

Stores a PTZ preset for the selected video panel. If the connected camera is a dome, the preset will be stored on the camera. If the connected camera is a fixed camera, the current digital zoom settings will be stored as preset on the decoder. If the connected camera is a Bosch/IQSIGHT panoramic camera, the current dewarping zoom settings will be stored as preset on the decoder. For digital and dewarping presets, the preset ID is restricted to [0..9].

```
"Output.StorePTZPreset.Request":
{
        "required": [ "output", "presetId" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "presetId": { "type": "integer", "minimum": 0, "maximum": 4095 }
        }
}

"Output.StorePTZPreset.Response": {}
```

Output Set PTZ Preset

Restores a PTZ preset for the selected video panel. If the connected camera is a dome, the preset will be restored by the camera. If the connected camera is a fixed camera, the previously stored digital zoom settings will be restored. If the connected camera is a Bosch/IQSIGHT panoramic camera, the previously stored dewarping zoom settings will be restored. For digital and dewarping presets, the preset ID is restricted to [0..9].

```
"Output.SetPTZPreset.Request":
{
        "required": [ "output", "presetId" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "presetId": { "type": "integer", "minimum": 0, "maximum": 4095 }
        }
}

"Output.SetPTZPreset.Response": {}
```

Output Set PTZ Speeds

Sets speed values for pan, tilt, zoom, iris, and focus for a PTZ-capable camera connected to the selected video panel. If the camera is not PTZ-capable, falls back to digital zoom or dewarping zoom.

```
"Output.SetPTZSpeeds.Request":
{
        "required": [ "output", "speeds" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "speeds": { "$ref": "#/definitions/PTZSpeedsData" }
        }
}
"Output.SetPTZSpeeds.Response": {}
```

Output Freeze

Freezes a selected video panel.

```
"Output.Freeze.Request":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" }
        }
}

"Output.Freeze.Response": {}
```

Output Un-Freeze

Un-freezes a selected video panel.

```
"Output.UnFreeze.Request":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" }
        }
}

"Output.UnFreeze.Response": {}
```

Line Enter Fullscreen

Enters an optionally selected fullscreen mode for a selected video panel. The picture-in-picture (PiP) fullscreen mode will display all other active video panels with reduced size on top of the maximized video panel.

```
"Line.EnterFullscreen.Request":
{
        "required": [ "output" ],
        "properties":
        {
                "output": { "$ref": "#/definitions/OutputData" },
                "mode": { "type": "string", "enum": [ "Single", "PiP" ], "default": "Single" }
        }
}

"Line.EnterFullscreen.Response": {}
```

Line Leave Fullscreen

Leaves the fullscreen mode for a selected line.

```
"Line.LeaveFullscreen.Request":
{
        "required": [ "lineId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 }
        }
}

"Line.LeaveFullscreen.Response": {}
```

Line Set Layout ID

Activates a selected video panel layout on a selected line.

```
"Line.SetLayoutId.Request":
{
        "required": [ "lineId", "layoutId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                "layoutId": { "type": "integer", "minimum": 0, "maximum": 2147483647 }
        }
}

"Line.SetLayoutId.Response": {}
```

Line Get Layout ID

Retrieves the identifier for the active layout of a selected line.

```
"Line.GetLayoutId.Request":
{
        "required": [ "lineId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 0, "maximum": 9 }
        }
}

"Line.GetLayoutId.Response":
{
        "required": [ "lineId", "layoutId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 0, "maximum": 9 },
                "layoutId": { "type": "integer", "minimum": 0, "maximum": 2147483647 }
```

```
        }
}
```

Line Get Layout

Retrieves a detailed video panel layout descriptor for a selected line and selected identifier of a supported layout on that line. Layout identifiers are specific for different line and screen configurations, supported layouts can be retrieved with the Lines.Get request.

```
"Line.GetLayout.Request":
{
        "required": [ "lineId", "layoutId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                "layoutId": { "type": "integer", "minimum": 0, "maximum": 2147483647 }
        }
}

"Line.GetLayout.Response":
{
        "required": [ "layout" ],
        "properties":
        {
                "layout": { "$ref": "#/definitions/LayoutData" }
        }
}
```

Line Get Coders

Retrieves detailed video panel information for a selected line.

```
"Line.GetCoders.Request":
{
        "required": [ "lineId" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 }
        }
}

"Line.GetCoders.Response":
{
        "required": [ "lineId", "coders" ],
        "properties":
        {
                "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                "coders":
                {
                        "type": "array",
                        "items":
                        {
                                "type": "object",
                                "required": [ "coderId", "tile" ],
                                "properties":
                                {
                                        "coderId": { "type": "integer", "minimum": 1, "maximum": 64 },
                                        "tile": { "$ref": "#/definitions/ScreenRectangleData" }
                                }
                        }
                }
        }
}
```

Lines Get

Retrieves detailed information about all available lines.

```
"Lines.Get.Request": {}

"Lines.Get.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "lineId", "screen", "refreshRateHz", "supportedLayoutIds" ],
                "properties":
                {
                        "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                        "screen": { "$ref": "#/definitions/ScreenRectangleData" },
                        "refreshRateHz": { "type": "integer", "minimum": 0, "maximum": 999 },
                        "supportedLayoutIds":
                        {
                                "type": "array",
                                "items": { "type": "integer", "minimum": 0, "maximum": 2147483647 }
                        }
                }
        }
}
```

Lines Freeze All

Freezes all video panels on all lines.

```
"Lines.FreezeAll.Request": {}

"Lines.FreezeAll.Response": {}
```

Lines Un-Freeze

Un-freeze all video panels on all lines.

```
"Lines.UnFreezeAll.Request": {}

"Lines.UnFreezeAll.Response": {}
```

Lines Store Favorite

Stores the current layout, active camera connections and active camera salvos for all lines as favorite on the decoder.

```
"Lines.StoreFavorite.Request":
{
        "required": [ "favoriteId" ],
        "properties":
        {
                "favoriteId": { "type": "integer", "minimum": 0, "maximum": 9 }
        }
}

"Lines.StoreFavorite.Response": {}
```

Lines Set Favorite

Restores for all lines a previously stored layout, including camera connections and camera salvos.

```
"Lines.SetFavorite.Request":
```

```
{
        "required": [ "favoriteId" ],
        "properties":
        {
                "favoriteId": { "type": "integer", "minimum": 0, "maximum": 9 }
        }
}

"Lines.SetFavorite.Response": {}
```

Banners Get Banner Salvo IDs

Retrieves information about the available banner salvo IDs, i.e. the IDs of all configured banner salvos.

```
"Banners.GetBannerSalvoIds.Request": {}

" Banners.GetBannerSalvoIds.Response":
{
        "type": "array",
        "items": { "type": "integer", "minimum": 1, "maximum": 16 }
}
```

Banners Get Banner Salvos

Retrieves the list of banner salvo assignments to certain locations given by line ID and screen position. A banner salvo ID with value 0 indicates that no banner is active at the given line and position.

```
"Banners.GetBannerSalvos.Request": {}

"Banners.GetBannerSalvos.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "lineId", "position", "bannerSalvoId" ],
                "properties":
                {
                        "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                        "position": { "type": "string", "enum": [ "Top", "Center", "Bottom" ],
"default": "Center" },
                        "bannerSalvoId": { "type": "integer", "minimum": 0, "maximum": 16 }
                }
        }
}
```

Banners Set Banner Salvos

Assigns one or more banner salvos to given locations defined by line ID and screen position. A banner salvo ID with value 0 indicates that no banner shall be active at the given line and position. Banner salvo locations not included in the list of banner salvo locations are not affected by the command and will keep their current banner salvo assignment.

```
"Banners.SetBannerSalvos.Request": {}

"Banners.SetBannerSalvos.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "lineId", "position", "bannerSalvoId" ],
                "properties":
```

```
                {
                        "lineId": { "type": "integer", "minimum": 1, "maximum": 9 },
                        "position": { "type": "string", "enum": [ "Top", "Center", "Bottom" ],
"default": "Center" },
                        "bannerSalvoId": { "type": "integer", "minimum": 0, "maximum": 16 }
                }
        }
}
```

Banners Stop All Banner Salvos

Stops all banner salvos and clears the banner salvo assignment for all lines and all screen positions.

```
"Banners.StopAllBannerSalvos.Request": {}

"Banners.StopAllBannerSalvos.Response": {}
```

Inputs Get

Retrieves details for all pre-configured video inputs (cameras). Pre-configuration is supported e.g. in the IQSIGHT Configuration Manager application's section for IP matrix setup. The input ID is then reflecting the PTZ keyboard's key code for the according camera.

```
"Inputs.Get.Request": {}

"Inputs.Get.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "inputId", "input" ],
                "properties":
                {
                        "inputId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                        "input": { "$ref": "#/definitions/InputData" },
                        "digitalZoom": { "$ref": "#/definitions/DigitalZoomData" },
                        "dewarpingZoom": { "$ref": "#/definitions/DewarpingZoomData" }
                }
        }
}
```

Outputs Get

Retrieves details for all pre-configured video panels. Pre-configuration is supported e.g. in the IQSIGHT Configuration Manager application's section for IP matrix setup. The output ID is then reflecting the PTZ keyboard's key code for the according video panel.

```
"Outputs.Get.Request": {}

"Outputs.Get.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "outputId", "output" ],
                "properties":
                {
                        "outputId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                        "output": { "$ref": "#/definitions/OutputData" }
                }
        }
```

```
}
```

Salvos Get

Retrieves details for all pre-configured video salvos. Pre-configuration is supported e.g. in the IQSIGHT Configuration Manager application's section for IP matrix setup. The salvo ID is then reflecting the PTZ keyboard's key code for the according video salvo.

```
"Salvos.Get.Request": {}

"Salvos.Get.Response":
{
        "type": "array",
        "items":
        {
                "type": "object",
                "required": [ "salvoId", "inputIds", "salvoCycleTimeS" ],
                "properties":
                {
                        "salvoId": { "type": "integer", "minimum": 0, "maximum": 4095 },
                        "inputIds":
                        {
                                "type": "array",
                                "items": { "type": "integer", "minimum": 0, "maximum": 4095 }
                        },
                        "salvoCycleTimeS": { "type": "integer", "minimum": 0, "maximum": 4294967295 }
                }
        }
}
```