Monitor Wall RCP+ Reference

Version 11.10

Contents

General Notes 7
Glossary 8
Transport Protocol 9
Remote Control Protocol Plus Header Layout 9
Packetizing in the TCP Stream 10
RCP+ Protocol Procedure 11
RCP+ over a HTTP Tunnel 12
RCP+ via CGI 13
Overview 13
Reading Parameters 13
Writing Parameters 15
CGI <<query_string>> reference 16
Mandatory CGI Parameters 16
Optional CGI Parameters 17
JSON RPC for Live Operations 18
Identification 19
HARDWARE_VERSION 19
SOFTWARE_VERSION 19
DEVICE_TYPE_IDS 20
SERIAL_NUMBER 20
UNIT_NAME 20
UNIT_ID 21
MONITOR_NAME 21
SUPPORTED_OPCODES 21
CAPABILITY_LIST 22
PRODUCT_NAME 23
COMMERCIAL_TYPE_NUMBER 23
System 24

APP_OPTION_UNIT_ID 24
APP_OPTION_SET 24
PERMISSIONS 24
SSD_CAPABILITIES 25
SSD_WRITE_PROTECTION_MODE 25
SSD_ENCRYPTION_MODE 26
USB_MODE 26
PTZIF_SENSITIVITY 27
OVERLOAD_STATUS 28
STARTUP_INFO 29
RUNTIME_INFO 29
HW_DECODING_DEFAULT_THRESHOLDS 29
HW_DECODING_THRESHOLDS_OVERRIDE 31
Network 33
MAC_ADDRESS 33
IP_STR 33
GATEWAY_IP_STR 33
SUBNET_STR 33
DEVICE_IP_ADDRESS 34
DHCP_VAL 34
DNS_SERVER_IP_STRING 35
HOST_NAME 35
LOCAL_HTTP_PORT 35
LOCAL_HTTPS_PORT 36
RCP_SERVER_PORT 36
MIN_TLS_VERSION 36
RTSP_PORT 36
RTSPS_PORT 37
Time 38
DATE_WDAY 38
DATE_DAY 38
DATE_MONTH 38
DATE_YEAR 38

TIME_HRS 39
TIME_MIN 39
TIME_SEC 39
TIMEZONE 39
TIME_ZONE_STRING 40
NTP_SERVER_IP_STR 40
FORCE_TIME_SET 40
SYSTEM_DATETIME_V2 40
Connection 41
PASSWORD_SETTINGS 41
ALARM_CONNECT_TO_IP_STR 41
REMOTE_PASSWORD 41
NBR_OF_ALTERNATIVE_ALARM_IPS 42
ALARM_CONNECT_TO_IP 42
CONNECT_TO 42
DISCONNECT_PRIMITIVE 44
ACTIVE_CONNECTION_LIST 45
CONNECT_URL 46
VERIFY_URL 47
FALLBACK_STREAMING_PROTOCOL 48
VIDEO_OUTPUT_STATUS 49
DISCOVERY_PORT 49
RECONNECT 50
VIDEO_MEMORY_BUDGET 51
VIDEO_MEMORY_LOAD 52
VIDEO_MEMORY_PROTECTION_STATE 52
Monitor 53
BOARD_RESET 53
DECODER_LAYOUT_LIST 53
DECODER_LAYOUT 55
CAMEO_DISTANCE 55
CAMEO_ALARM_MODE 56
CAMEO_ALARM_FRAME_COLOR 56

CAMEO_ALARM 57
SWAP_MONITOR 57
SHOW_METADATA 59
SHOW_METADATA2 59
OVERLAY_MODE 60
INSIGHTS_OVERLAY_MODE 60
AUTO_RESOLUTION_CHANGE 60
TARGET_DISPLAY_ORIENTATION 61
SMOOTHNESS 62
RENDERING_MODE 62
PREFERRED_ASPECT_RATIO 63
NUMBER_OF_DECODERS 64
ZOOM_MODE 64
DIGITAL_ZOOM_PERSISTENCE_MODE 64
IN_WIN_PTZ_MODE 66
CONTEXT_MENU_MODE 66
MONITOR_INFO 67
MONITOR_INFO2 68
MONITOR_POWER_MODE 68
VOCS_CONFIG 69
VOCS_CONFIG2 69
Snapshot download 71
Background logo up- and download 73
LOGO_LIST 73
Banners 74
Banner up- and download 74
BANNER_LIST 74
CONFIG_BANNER_SALVO 75
CONFIG_BANNER_SALVOS 75
CONNECT_BANNER_SALVO 76
Stamping 78
NAME_STAMP_VAL 78
TIME_STAMP_VAL 78

ALARM_DISP_VAL 78
STAMP_ATTR_NAME 79
STAMP_ATTR_TIME 80
STAMP_ATTR_ALARM 80
ALARM_STRING 80
Keyboard and IP Matrix Configuration 81
KBD_CONFIG_CAMERA_STR 81
KBD_CONFIG_MONITOR_STR 82
KBD_CONNECT_PARAMS 82
KBD_PASSWORD_CAMERA 83
KBD_PASSWORD 83
KBD_TIMEOUT 83
KBD_CONFIG_SALVO 83
KBD_CONFIG_SALVO2 84
KBD_CONFIG 85
DECODER_GROUP 86
FREEZE_CHECK_INTERVAL 87
SALVO_PAUSE_TIMEOUT 87
KBD_CONFIG_CAMERA_MAPPING 87
Keyboard and IP Matrix Operations 89
INFO_OVERLAY_MODE 89
KBD_CONNECT_CAMERA 89
KBD_CONNECT_SALVO 89
KBD_GOTO_PRESET 90
KBD_STORE_PRESET 91
KBD_LAYOUT_STEP 91
PTZIF 92
FULLSCREEN_TILE 93
FREEZE_MODE 94
SALVO_PAUSE_STATE 95
ACTIVE_TILE 95
SAVE_FAVORITE 96
LOAD_FAVORITE 96

KBD_LICENSED_CHANNELS 96
KBD_LICENSED_CHANNELS_GROUP 96
SERIAL_PORT_APP_VAL 97
TRANSFER_TRANSPARENT_DATA 97
Configuration Sealing 99
CONFIG_SEALING_ENABLED 99
CONFIG_SEALING_STATUS 99

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.

Trademarks

All names used in this manual for hardware and software are very probably registered trademarks 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 (tile) can display one video stream.
Layout	Arrangement of video panels (coders) on a screen (line). Each screen can have its own layout.
IP Matrix	Cluster of 1 to 4 hardware decoders that control up to 16 physical screens.
VOCS	Visual output capture service

Transport Protocol

The transport protocol for remote control of the Monitor Wall application follows the general specifications for Bosch/IQSIGHT Video over IP cameras. For further details, please refer to the corresponding Rcp+ specification document. This section reproduces some basic chapters from the camera's Rcp+ specification.

Remote Control Protocol Plus Header Layout

The RCP Plus protocol header consists of 16 Bytes. The begin of the payload section is on DWORD boundary for version 3.

<table>
<tr><td colspan="4"></td><td colspan="2">16</td><td colspan="2">32</td></tr>
<tr><td colspan="4">Tag
2 Bytes</td><td>Data Type
1 Byte</td><td>Version
4 Bits</td><td>R/W
4 Bits</td></tr>
<tr><td>C
1</td><td>T
1</td><td>Action
6 Bits</td><td>Reserved
1 Byte</td><td colspan="3">Client ID
2 Bytes</td></tr>
<tr><td colspan="7">Session ID
4 Bytes</td></tr>
<tr><td colspan="4">Numeric Descriptor
2 Bytes</td><td colspan="3">Payload Length
2 Bytes</td></tr>
</table>

Tag

Each tag is represented by two octets. It identifies the command which should be processed by the Videojet.

Data Type

Specifies the data type of the payload section. These are the currently available data types:

Values:

F_FLAG	0x00 (1 Byte)
T_OCTET	0x01 (1 Byte)
T_WORD	0x02 (2 Byte)
T_INT	0x04 (4 Byte)
T_DWORD	0x08 (4 Byte)
P_OCTET	0x0C (N Byte)
P_STRING	0x10 (N Byte)
P_UNICODE	0x14 (N Byte)

Version

The current RCP version is 3. Backward compatibility to version 2 or version 0 is NOT provided.

R/W

Specifies whether the command should read or write. The Read/Write field is coded in the lower nibble of byte 4.

Values:

Read	0x0
Write	0x1

T

String table available. This bit signals, when set, that there is a string table appended to this RCP+ packet which contains IPv6 addresses or host names.

C
Continuation. This bit signals, when set, that this RCP+ packet is not terminated in the payload; additional packets with the full RCP+ header will follow immediately in the stream as long as this bit is cleared. The reassembly of the complete payload is up to the application and is beyond the scope of this document.

Action
Specifies the kind of the packet.

Values:

Request	0x00
Reply	0x01
Message	0x02
Error	0x03

Client ID
Each RCP client register results in a Client ID; this ID has to be provided in all following RCP commands.

Session ID
This ID is used for implementations which need to identify a once registered user in other applications or RCP sessions.

Numeric Descriptor
The Numeric Descriptor specifies an attribute for components which are installed more than one time inside the Videojet, e.g. inputs or relays. The first component is always counted as 1. If this field is not applicable to the command in this packet, a value of zero should be inserted.

Payload Length
The number of data bytes inside the payload section. The length field itself is not counted.

Reserved
This byte is returned by the Videojet unchanged. It is up to the user to setup a request ID here to assign the replies to multiple pending requests.

Packetizing in the TCP Stream
As TCP is a stream-oriented protocol, packetizing information has to be inserted to achieve proper packet reconstruction. For this purpose the TPKT structure is used.

16 32

Version 1 Byte	**Reserved** 1 Byte	**Length** 2 Bytes
RCP Packet N Bytes		

8 24

Version

TPKT version 3.

Reserved

Should be set to zero.

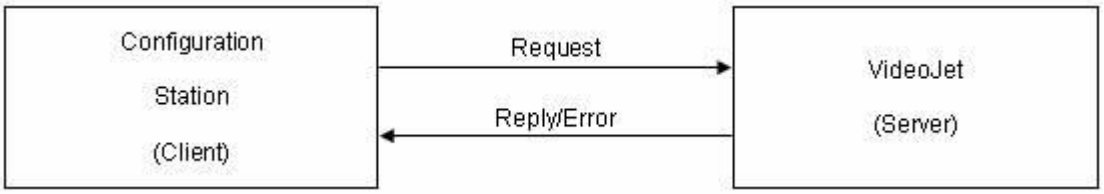
Length

Byte count of the complete packet; RCP packet length + TPKT length (4 Bytes).

RCP+ Protocol Procedure

Request / Reply

The VideoJet unit will send back a reply on each incoming request. In some protocol functions the reply will carry no data.



NOTE: A reply from a write command which is not readable will be an RCP_ERROR_READ_NOT_SUPPORTED (error code 0x90) This means that it is not possible to determine correct setting of the command.

Messages

A message will be generated on the VideoJet on certain events (see RCP command for details). If an RCP client is registered to receive a certain message (e.g. an input state has changed), the VideoJet will generate it. The message has the same payload format as the response to a Read command of the same tag, except the reserved byte. The reserved byte is used as sequence number.



RCP is capable of sending messages on certain events. Before an RCP client can receive these messages, a registration at the VideoJet is necessary.

RCP Errors

The packet will have the standard layout with the method field set to 'Error'. The first bytes of the payload section contains the error cause. If the error code is RCP_ERROR_COMMAND_SPECIFIC, then the command specific error (see RCP command for details) is included in the second byte. The following generic error codes are defined:

RCP_ERROR_UNKNOWN	0xFF
RCP_ERROR_INVALID_VERSION	0x10
RCP_ERROR_NOT_REGISTERED	0x20
RCP_ERROR_INVALID_CLIENT_ID	0x21
RCP_ERROR_INVALID_METHOD	0x30
RCP_ERROR_INVALID_CMD	0x40
RCP_ERROR_INVALID_ACCESS_TYPE	0x50
RCP_ERROR_INVALID_DATA_TYPE	0x60
RCP_ERROR_WRITE_ERROR	0x70
RCP_ERROR_PACKET_SIZE	0x80
RCP_ERROR_READ_NOT_SUPPORTED	0x90

RCP_ERROR_INVALID_AUTH_LEVEL	0xa0
RCP_ERROR_INVAILD_SESSION_ID	0xb0
RCP_ERROR_TRY_LATER	0xc0
RCP_ERROR_TIMEOUT	0xd0
RCP_ERROR_NO_LICENCE	0xe0
RCP_ERROR_COMMAND_SPECIFIC	0xf0
RCP_ERROR_ADDRESS_FORMAT	0xf1

Note the error code 0xc0 RCP_ERROR_TRY_LATER indicates, that the VideoJet recognizes the command, but it cannot be processed immediately. The client should repeat this command later.

RCP+ over a HTTP Tunnel

Many network installations include firewalls to protect network segments against unauthorized access. In many cases, the HTTP port (80) is enabled by default to gain Internet access. For this purpose, RCP offers a second network port. As the VideoJet has a built-in webserver, the connection can be made to the webserver, the token

`GET /rcp_tunnel HTTP1.0\r\n\r\n`

advises the webserver to pass this socket to the RCP server. From that point on, the client is connected to the RCP server and can immediately continue with RCP registration.

Note: the delimiters '\r' and '\n' are written here in C notation, in the TCP packet to the server these delimiters correspond with the ASCII codes 0x0d and 0x0a.

Example:

1. Establish a TCP connection to port 80 at your Videojet
2. Send the string `GET /rcp_tunnel HTTP/1.0\r\n\r\n` (as HEX string: 0x47 0x45 0x54 0x20 0x2f 0x72 0x63 0x70 0x5f 0x74 0x75 0x6e 0x6e 0x65 0x6c 0x20 0x48 0x54 0x54 0x50 0x31 0x2e 0x30 0x0d 0x0a 0x0d 0x0a)
3. Proceed in the same manner as for a normal RCP connection

RCP+ via CGI

Overview

Bosch/IQSIGHT video over IP products in video management system environments are mainly controlled and managed using an enhanced version of Remote Control Protocol, RCP+, pronounced »RCP plus«. This protocol defines commands and messages that allow to configure the units and to establish communication between units, or units and management systems.

RCP+ commands can also be encapsulated into CGI requests. The command structure of RCP+ must be translated into parameters and values handed over to the unit's XML-based CGI interpreter as a query string. Using HTTP as transportation protocol, e.g. with a web browser, an URL may look like:

```
http://192.168.0.1/rcp.xml?<<query_string>>
```

Each RCP+ command is a fixed predefined 16 bit value followed by a structured set of parameters, some mandatory, some optional. All message elements in the query string must resemble an appropriate pair of parameter and value. The following examples shall give a principle overview of how to use RCP+ commands with CGI.

Reading Parameters

Various parameters can be read from a Bosch/IQSIGHT video over IP product, e.g. the hardware or firmware version, or the name of the unit. To get e.g. the serial number (SERIAL_NUMBER), the URL must look like:

```
http://192.168.0.1/rcp.xml?command=0x0ae7&type=P_STRING&direction=READ
```

The web browser is able to display the returned XML message in a readable format.

The XML return message is displayed in the web browser like this (the unit's fake serial number is 0123456789ABCDEF):

```
<rcp>
      <command>
            <hex>0x0ae7</hex>
            <dec>2791</dec>
      </command>
      <type>P_STRING</type>
      <direction>READ</direction>
      <num>0</num>
      <idstring>0</idstring>
      <sessionid>0x00000000</sessionid>
      <auth>0</auth>
      <result>
            <str>0123456789ABCDEF</str>
      </result>
</rcp>
```

The return message displayed in the web browser also contains unused and optional parameters that may be represented by an »empty« XML tag.

Here is another example that shows the response structure for simple numbers like an unsigned short integer (`T_WORD`). Http port number request (LOCAL_HTTP_PORT):

httphttp://192.168.0.1/rcp.xml?command=0x0954&type=T_WORD&direction=READ

Response:

```
<rcp>
      <command>
            <hex>0x0954</hex>
            <dec>2388</dec>
      </command>
      <type>T_WORD</type>
      <direction>READ</direction>
      <num>0</num>
      <idstring>0</idstring>
      <sessionid>0x00000000</sessionid>
      <auth>0</auth>
      <result>
            <hex>0x50</hex>
            <dec>80</dec>
      </result>
</rcp>
```

Commands with structured payloads like the decoder layout list (DECODER_LAYOUT_LIST) all have the P_OCTET type:

```
http://192.168.0.1/rcp.xml?command=0x09a1&type=P_OCTET&direction=READ
```

Such commands return the byte arrays as sequence of space-separated hex numbers, with one two-digit number per byte:

```
<rcp>
      <command>
            <hex>0x09a1</hex>
            <dec>2465</dec>
      </command>
      <type>P_OCTET</type>
      <direction>READ</direction>
      <num>0</num>
      <idstring>0</idstring>
      <sessionid>0x00000000</sessionid>
      <auth>0</auth>
      <result>
            <str>00 01 00 02 00 03 00 04 10 05 02 51 02 71 12 91 12 b1 12
      d1 12 f1 </str>
      </result>
</rcp>
```

Writing Parameters

RCP+ over CGI can also be used to change the various parameters of a Bosch/IQSIGHT IP Video product. Parameter writing or setting is done in a similar way like reading them, except the direction tag is set to `WRITE` instead of `READ` and a payload has to be specified. For example, to change the distance between the video panels (CAMEO_DISTANCE) the URL must look like:

```
http://192.168.0.1/rcp.xml?command=0xe209&type=F_FLAG&direction=WRITE&payload=0x2
```

The XML return message is displayed in the web browser like this:

```
<rcp>
	<command>
		<hex>0xe209</hex>
		<dec>57865</dec>
	</command>
	<type>T_INT</type>
	<direction>WRITE</direction>
	<num>0</num>
	<idstring>0</idstring>
	<sessionid>0x00000000</sessionid>
	<auth>0</auth>
	<result>
		<hex>0x2</hex>
		<dec>2</dec>
	</result>
</rcp>
```

Some commands may return the set value in the payload.

Here is an example for a write command with a structured payload (CAMEO_ALARM):

```
http://127.0.0.1/rcp.xml?command=0xe268&type=P_OCTET&direction=WRITE&payload=0x01020008
```

The payload consists of one byte for the line number (screen, display), one byte for the coder number (video panel, tile), and two bytes (unsigned short integer, network byte order) for the cameo alarm duration in seconds. The response looks like this, no payload is returned:

```
<rcp>
	<command>
		<hex>0xe268</hex>
		<dec>57960</dec>
	</command>
	<type>P_OCTET</type>
	<direction>WRITE</direction>
	<num>0</num>
	<idstring>0</idstring>
	<sessionid>0x00000000</sessionid>
	<auth>0</auth>
```

```
      <result>
            <str/>
      </result>
</rcp>
```

CGI <<query_string>> reference

As described and shown in the previous section, the CGI query string consists of different parameter-value pairs. The parameters and their possible values are described in the following sections. Some of them are mandatory, some of them are optional. First the mandatory parameters are described, followed by the optional ones.

Mandatory CGI Parameters

Parameter »command«

This is the most important part of the CGI request and represents the RCP+ command tag.

Values:

RCP+ command tag value.

Parameter »type«

The payload type for a specific RCP+ command tag.

Values:

```
F_FLAG
T_OCTET
T_WORD
T_INT
T_DWORD
P_OCTET
P_STRING
P_UNICODE
```

Parameter »direction«

Defines, whether the call is getting or setting the parameter in question.

Values:

```
READ
WRITE
```

Parameter »num«

The numeric descriptor parameter. This parameter is mandatory for certain RCP+ commands.

Values:

The numeric descriptor.

Parameter »payload«

The payload for a specific RCP+ command. The payload structure is depending on the command's payload type as described above.

Values:

The payload as readable string for payload type `P_STRING`

The payload as octet array with no spaces preceded with 0x for type `P_OCTET` and `P_UNICODE`

The payload value in hex or decimal notion for all other payload types.

Optional CGI Parameters

Parameter »idstring«

Not supported.

Parameter »sessionid«

Values:

Context specific session ID for certain commands

JSON RPC for Live Operations

The Monitor Wall application supports JSON remote procedure commands (RPC) for certain live operations like changing the layout and connecting cameras. The JSON RPC schema can be downloaded from the Monitor Wall's built-in web server. For further details please refer to the separate JSON RPC documentation, also downloadable from the built-in web server.

Identification

HARDWARE_VERSION

	Tag code	NumDes	Message	SNMP Support
	0x002e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the hardware version as hex string	
Write			not supported	

or

	Tag code	NumDes	Message	SNMP Support
	0x002e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	noprot	read the hardware version as DWORD	
Write			not supported	

Gets the system hardware version.

f1004940	VIP XD HD (discontinued)
f1005440	VJD7000 (discontinued)
f1009040	Videojet decoder family (VJD-8000, VJD-7513, VJD-7523, VJD-7533), see DEVICE_TYPE_IDS
f100fe40	Monitor Wall stand-alone software

Rcp+ via CGI read:

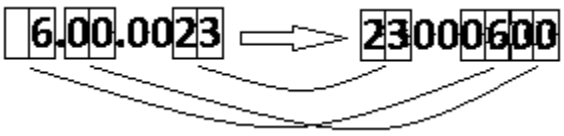
```
/rcp.xml?command=0x002e&type=T_DWORD&direction=READ
```

SOFTWARE_VERSION

	Tag code	NumDes	Message	SNMP Support
	0x002f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the software version as hex string	
Write			not supported	

Gets the system software version.

For example: For build 6.00.0023 result is 23000600:



Rcp+ via CGI read:

```
/rcp.xml?command=0x002f&type=P_STRING&direction=READ
```

DEVICE_TYPE_IDS

	Tag code	NumDes	Message	SNMP Support
	0x0b07	no	No	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	read the device type ids	
Write			not supported	

Gets the device type IDs.

1st four bytes: Product ID

- 0x49 VIP XD HD (discontinued)
- 0x54 VJD-7000 (discontinued)
- 0x90 Videojet decoder family (VJD-8000 (discontinued), VJD-75x3)
- 0xfe Monitor Wall stand-alone software

2nd four bytes: Variant ID (supported only for Videojet decoder family)

- 0 VJD-8000 (discontinued)
- 1 VJD-7513
- 2 VJD-7523
- 3 VJD-7533
- 4 VJD-7543

3rd four bytes: reserved

SERIAL_NUMBER

	Tag code	NumDes	Message	SNMP Support
	0x0ae7	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	returns the serial number of the device	
Write			not supported	

Gets the system serial number.

If it is not defined the result is “00000000000000000”. Max length is 127characters.

UNIT_NAME

	Tag code	NumDes	Message	SNMP Support
	0x0024	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	user	read the unit name	
Write	p_unicode	service	set unit name	

Specifies the unit name(max 32 unicode characters).

UNIT_ID

	Tag code	NumDes	Message	SNMP Support
	0x0025	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	user	read the unit ID	
Write	p_unicode	service	set unit ID	

Specifies the unit id (max 32 unicode characters).

MONITOR_NAME

	Tag code	NumDes	Message	SNMP Support
	0x028a	line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	user	read the video monitor name	
Write	p_unicode	service	set the video monitor name (32 characters max)	

Gets the name of each line. Default return value: “Monitor <Line> (<IpAddress>)”

SUPPORTED_OPCODES

	Tag code	NumDes	Message	SNMP Support
	0xe235	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	list of supported opcodes for this session	
Write			not supported	

Gets the list of supported opcodes for this session.

Reply payload Structure

OpCode 1 4 Byte
... 4 Bytes
OpCode N 4 Byte

CAPABILITY_LIST

	Tag code	NumDes	Message	SNMP Support
	0xff10	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	see detailed description	
Write			not supported	

Gets the list of system capabilities.

Reply payload Structure

0xBABA 2 Bytes	Version 2 Bytes	NbSection 2 Bytes	Section 1	…	Section N

Version

current version of the capabilities (0x0001)

NbSection

Number of following sections. Hardcoded (0x0019)

Section Structure

Type 2 Bytes	Size 2 Bytes	NbElement 2 Bytes	Element 1	…	Element N

Type

Type of Element

Values:

Video	0x0001

Size

Size of the section including SectionType, Size and NbElement. If the section is unknown, you can skip to the next section using the size.

NbElement

Determines how many Elements are following. The definition of each Element depends on the type of the section.

Element Structure

Type 2 Bytes	Identifier 2 Bytes	Compr 2 Bytes	InputNo 2 Bytes	Resolution 2 Bytes

Type

is one of the following

Values:

VIDEO_ENCODER (VOCS)	0x0001
VIDEO_DECODER	0x0002

Identifier

One-based coder number.

Compression

is one or multiple of the following

Values:

VIDEO_COMP_H264	0x0004
VIDEO_COMP_JPEG	0x0008
VIDEO_COMP_H265	0x0010

InputNo

One-based line number.

Resolution

Not used.

Other sections are not supported by Monitor Wall. The number of decoders can be influenced by the NUMBER_OF_DECODERS command.

PRODUCT_NAME

	Tag code	NumDes	Message	SNMP Support
	0x0aea	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the commercial type number (CTN) of the device (SKU number).	
Write			not supported	

COMMERCIAL_TYPE_NUMBER

	Tag code	NumDes	Message	SNMP Support
	0x0be7	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the commercial type number (CTN) of the device (SKU number).	
Write			not supported	

System

APP_OPTION_UNIT_ID

	Tag code	NumDes	Message	SNMP Support
	0x09e1	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	read unique unit id (installation code)	
Write			not supported	

Read out the unique unit id (installation code) for setting application options (licenses keys).

APP_OPTION_SET

	Tag code	NumDes	Message	SNMP Support
	0x09e2	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	read installed application options	
Write	p_string	service	install application options [automatic application restart on change]	

Read request returns a readable string with details on all installed application options (license type, count, authorization number, start and end date if applicable).
Write request either installs an application option when the payload represents a license key, or uninstalls an application option when the payload represents the authorization number (order number or purchase number) of an previously installed application option. A write request response does not contain a payload, the check for a successful installation of an application option requires an inspection of the read request response after the device's application restart (triggered automatically by the write request).
Bosch/IQSIGHT hardware decoders may include pre-installed application options that cannot be deleted, identifiable by authorization numbers like "BoschDevice" and "IQSIGHTDevice".

PERMISSIONS

	Tag code	NumDes	Message	SNMP Support
	0xe22a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	read permissions	
Write	t_int	service	write permissions	

Gets and sets the permissions as bit-field (only the lower 16 bit are used):
Bit 0 (0x1): [Discontinued] If set, permission to access the recording is granted (replay permission)

For write access, the upper 16 bit can optionally be used as mask field. If a mask is provided, only those permission flags will be changed where a mask bit is set. An empty mask (value 0x0000) will be interpreted like a full mask (0xFFFF).

SSD_CAPABILITIES

	Tag code	NumDes	Message	SNMP Support
	0xe226	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	user (VJD only)	read the SSD capabilities	
Write			not supported	

Gets the SSD capabilities as bit-field:

Bit 0 (0x1): Write protection feature is supported

Bit 1 (0x2): SSD encryption supported in principle, but write protection is turned on and blocks changes to the SSD encryption mode

Bit 2 (0x4): SSD encryption supported

SSD_WRITE_PROTECTION_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe227	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user (VJD only)	read the SSD write protection mode	
Write	t_dword	service (VJD only)	set the SSD write protection mode [automatic device reboot on change]	

Gets/sets the SSD write protection mode:

0: Off

1: On

Only supported when the SSD write protection feature is supported.

The device will perform an immediate automatic reboot after a change of the SSD write protection mode.

Note: If SSD write protection feature is supported but SSD write protection is turned off, the device will enter an extended configuration mode. In this mode, extensive changes to configuration can be performed, like SSD encryption. Standard configuration is still possible in extended configuration mode, but video connections are not supported. The user can distinguish this mode by the extended configuration mode icons shown in the middle of each video panel. By turning on the SSD write protection, the device will leave the extended configuration mode and will resume normal operation after the automatic reboot.

SSD_ENCRYPTION_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe228	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	user (VJD only)	read the SSD encryption mode	
Write	t_word	service (VJD only)	set the SSD encryption mode [automatic device reboot on change]	

Gets/sets the SSD encryption mode:

0: Off

1: On

Can only be set when the SSD write protection feature is either not supported or turned off (see SSD_CAPABILITIES and SSD_WRITE_PROTECTION_MODE).

The device will perform an immediate automatic reboot after a change of the SSD encryption mode.

Encryption/decryption of the SSD may take several minutes, a progress indicator is shown on the screen.

The monitor wall service is not available while encryption/decryption is still running.

Attention: As long as the SSD is encrypted, the factory reset feature is not available. Before performing a factory reset, turn off SSD encryption.

USB_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe229	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	user (VJD only)	read the USB lockdown mode	
Write	t_word	service (VJD only)	set the USB lockdown mode	

Gets/sets the USB lockdown mode:

0: Off, all USB human interface devices (mouse, keyboard) are supported

1: On, USB is locked down, no USB devices are supported

USB lockdown is only active when the SSD write protection is turned on (see SSD_WRITE_PROTECTION_MODE).

Attention: After turning on the USB lockdown, no human interface devices are supported, thus the serviceability of the device is severely restricted. The only remaining access method is the Ethernet connector.

PTZIF_SENSITIVITY

	Tag code	NumDes	Message	SNMP Support
	0xe247	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	read PTZIF sensitivity	
Write	p_octet	service (VJD only)	set PTZIF sensitivity	

The PTZIF sensitivity determines the relationship between the analogue PTZ keyboard joystick deflection and the actual speed values that are used for changing pan, tilt, zoom, iris and focus settings for the camera on the active video panel. The sensitivities for the five speeds can be configured independently. The unit of the sensitivity values is percent. The default sensitivity is 100.

Each sensitivity value can be chosen as a multiple of 10 in the interval [-200,200]. Values that deviate from that scheme will be adapted or corrected as appropriate. Negative values change the direction, e.g. jet plane pilots may want to use negative values for the tilt sensitivity, which effectively inverts the Y-axis of the joystick.

Absolute sensitivity values larger than 100 do not extend the maximum supported speed for that axis, the maximum speed will just be reached earlier at a smaller deflection of the joystick. Absolute sensitivity values lower than 100 restrict the maximum achievable speed on the according axis for a full joystick deflection.

If all five sensitivities shall have the same value, the write payload may be set to just one 4 byte signed integer. If at least one of the pan, tilt, zoom, iris or focus sensitivities deviates from the others, the write payload must consist of a sequence of five 4 byte signed integers, with the sensitivity value order given by pan (first), tilt, zoom, iris, and focus (last), analogue to the capital letter sequence in the command name “PTZIF”. The read payload always contains five 4 byte integers, even if all of them have the same value.

The PTZ keyboard control of one or more of pan, tilt, zoom, iris and focus can be disabled by setting the according sensitivity value to zero.

OVERLOAD_STATUS

	Tag code	NumDes	Message	SNMP Support
	0xe248	no	yes	yes
	Datatype	**Access Level**	**Description**	
Read	t_dword	user	read or get notification on overload status	
Write	t_dword	service	emulate an overload status to trigger SNMP events	

The overload status is the basis for SNMP support. The payload consists of a bit field that indicates overload states. Multiple overload states can be active in parallel. The following overload states can be indicated:

0x1
Processor overload, device is in I-frame-only decoding mode since CPU/GPU cannot sustain decoding and rendering of all incoming frames; reduce number of connected streams to recover normal operation, or (if applicable) restrict the rendering frame rate via RENDERING_MODE.

0x2
Memory overload, device RAM is close to its critical limit, but device remains 100% operational; in case that the memory consumptions increases further, the device will reboot automatically when the critical limit is reached; memory overload may get caused by an excessive number (beyond the device's specification) of connected high resolution video streams.

0x4
Thermal overload, device is in I-frame-only decoding mode since the CPU/GPU temperature exceeds a certain first self-protection limit; device will recover as soon as the temperature drops; if the device continues to heat up nonetheless, e.g. due to an unfavorable environment, the device will automatically shut down once the CPU/GPU temperature exceeds a certain second self-protection limit; to avoid future thermal overload situations, increase the airflow around the device to improve heat dissipation, reduce number of connected streams, or (if applicable) restrict the rendering frame rate via RENDERING_MODE.

STARTUP_INFO

	Tag code	NumDes	Message	SNMP Support
	0xe262	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	service	read startup info as multiline string	
Write			not supported	

The startup info provides detailed insights into the system, compiled once after MonitorWall application start.

RUNTIME_INFO

	Tag code	NumDes	Message	SNMP Support
	0xe263	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	service	read runtime info as multiline string	
Write			not supported	

The runtime info provides detailed insights into the system, compiled every time anew.

HW_DECODING_DEFAULT_THRESHOLDS

	Tag code	NumDes	Message	SNMP Support
	0xe25d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (MW only)	read default threshold values	
Write			not supported	

The Monitor Wall application supports always CPU-based decoding. In order to optimize the decoding performance, the workstation's power mode should be configured to the highest available performance level in the according Windows control panel. Depending on individual video stream characteristics, the Monitor Wall application selects single-threaded or multi-threaded decoding.

For some workstations, the Monitor Wall application supports outsourcing of video decoding to hardware accelerators on the GPU. This is implemented for most Intel on-board GPUs and most Nvidia graphics cards, and is tested on a certain set of target platforms before each release. Please note that GPU-based decoding is not supported in mixed setups, i.e. when an Intel on-board GPU is active in a workstation that hosts an active Nvidia graphics card. In such cases, disabling Intel's on-board GPU in Window's device manager is an option to at least enable GPU-based on the Nvidia graphics card.

The Monitor Wall application features a heuristic that decides whether a video stream is decoded on the CPU or on the GPU, depending on individual video stream characteristics and GPU characteristics. A GPU may have a limit for the number of parallel hardware decoder instances (true for most Nvidia GPUs), and does have a limit for the overall decoding power. In general, the following guidelines are used by the heuristic:

- Streams with sub-HD resolution shall be decoded on the CPU

- The maximum supported number of hardware decoder instances on the GPU shall not be exceeded
- The maximum supported total decoding load on the GPU shall not be exceeded

Given the huge variance in possible video stream characteristics, the actual decoding load of a video stream on the GPU is hard to predict. The Monitor Wall application uses a metric for each video stream to estimate the decoding load before selecting a decoder on the CPU or GPU, using only video stream characteristics that can be easily extracted from the first key frame before decoding. This boils down to video width and height, since the bit rate of the encoded stream and the frame rate cannot be derived from the first encoded frame. Assuming a frame rate of 30fps, the pixel rate for the decoded stream can be calculated. This pixel rate is used as metric for the decoding load.

The Monitor Wall application features an internal database, where the overall decoding power for various GPU architectures and models is listed as overall pixel rate. These values have been determined with dedicated measurements using video streams with various codecs, resolutions, bit rates, and frame rates.

This approach works reasonably well for video streams featuring a frame rate of 30fps, and errs on the safe side for video streams with lower frame rates. Erring on the safe side means switching too early from GPU-decoding to CPU-decoding, i.e. the GPU is then not fully loaded when the application switches to CPU-decoding. With the upcoming video streams that feature 60fps, this metric breaks since the estimated decoding load is too small, because 30fps is assumed in the metric's formula. When connecting many 60fps streams with resolutions full HD and higher, the GPU will get overloaded by the decoding before the maximum number of parallel hardware decoder instances is reached. This will not damage the hardware, but decoding will be slowed down, real-time decoding is no longer possible. The rendered video streams will reflect this overload on the monitors by a continuous switching between building up a video delay and dropping video intervals to catch up.

The command HW_DECODING_DEFAULT_THRESHOLDS provides insights on the actual decision criteria for each individual workstation hardware. The values will be transmitted as sequence of tag/value pairs:

Tag	**Value**
2 Bytes	4 Bytes

The tags are two byte unsigned integers, the values are four bytes signed integers.

Tag	Value description
1	Maximum number of parallel hardware decoder instances for H.264
2	Maximum number of parallel hardware decoder instances for H.265
3	Maximum number of parallel hardware decoder instances, codec-independent
4	Maximum supported overall pixel rate for H.264 hardware decoding, in megapixels per second
5	Maximum supported overall pixel rate for H.265 hardware decoding, in megapixels per second
6	Maximum supported overall pixel rate for any hardware decoding, in megapixels per second
7	Minimum required number of pixels per frame for H.264 hardware decoding, in kilopixel
8	Minimum required number of pixels per frame for H.265 hardware decoding, in kilopixel
9	Assumed video frame rate for video decoding load prediction (30fps by default)

HW_DECODING_THRESHOLDS_OVERRIDE

	Tag code	NumDes	Message	SNMP Support
	0xe25e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (MW only)	read thresholds override values	
Write	p_octet	service (MW only)	write thresholds override values [automatic application restart on change]	

The default behavior of the heuristic that decides whether a video stream will be decoded on the GPU or on the CPU is not suitable for all conceivable combinations of video stream characteristics and GPU characteristics. Fine-tuning of the heuristic is possible with this command to optimize the usage of GPU hardware acceleration in the Monitor Wall application. Furthermore, field issues with untested GPUs (e.g. newly released GPUs, newly released GPU drivers) may cause issues with the hardware decoding feature. This command is also suitable to deactivate GPU-decoding for one or all codecs.

Note: On a Videojet decoder device, this fine-tuning is performed before product release, and cannot be altered with this command.

The payload structure for a read command mirrors the payload structure of HW_DECODING_DEFAULT_THRESHOLDS. By default, no threshold is overridden, and the read command will return -1 for each value.

The payload structure for a write command is a sequence of those tag/value pairs that shall be changed, in any order and any number. The provided values may be clipped by the Monitor Wall application. Setting a value to -1 clears the according threshold override.

Fine-tuning becomes necessary in the following cases:

- Some video panels do not show video, dependent on the order in which the video streams were connected. If always the first few connected video panels work well, and the next few video panels don’t work, and if it then continues to work well when the number of connected streams exceeds the values for tags 1, 2 or 3, then the threshold for the decision to switch to CPU-decoding is too high. The used GPU model supports less hardware decoding instances in parallel than listed in the internal database for this GPU’s architecture. Reducing the values for tags 1, 2 and 3 should fix this issue.
- CPU overload occurs sometimes while GPU-decoding load is low, depending on the order in which the streams were connected. This may happen when there are video streams with very low resolutions connected next to video streams with very high resolutions. When connecting the low-res streams first, the available limited GPU-decoding instances will be occupied with low-res video streams that have a low video decoding load on the GPU. Once all GPU-decoding instances are occupied, any subsequentially connected high-res video streams will be decoded on the CPU, causing significant stress on the CPU. This effect can be mitigated by increasing tag values 7 and 8, which will prevent the occupation of GPU-decoding instances with low-res video streams. This may need to be done iteratively, too high values for tags 7 and 8 may also lead to a higher load on the

CPU, depending on how many streams with video resolutions below the pixels-per-frame threshold will be connected.

- GPU-decoding overload occurs before the maximum number of GPU-decoding instances is reached. This may happen e.g. when many video streams with full HD resolutions (or higher) and with frame rates beyond 30fps are connected, or when the actual GPU model deviates from the expected behavior for the GPU's architecture. This can be mitigated in two ways:
    - The assumed video frame rate for video load prediction can be increased (e.g. to 60fps) via tag 9, which makes sense if most connected video streams will feature 60fps. If the increase of tag 9's value is not sufficient to resolve the issue, the values for tags 4, 5 and 6 can be reduced to force an earlier switching to CPU-decoding.
    - If the GPU-decoding overload issue is not related to the video stream frame rate, a general reduction of the values for tags 1, 2 and 3 should fix the issue.

Network

MAC_ADDRESS

	Tag code	NumDes	Message	SNMP Support
	0x00bc	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	read out the systems MAC address	
Write			not supported	

Gets the system MAC address.

IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x007c	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the unit's IP address using string notation (xxx.xxx.xxx.xxx)	
Write	p_string	service (VJD only)	set unit's IP address using string notation (xxx.xxx.xxx.xxx) [automatic application restart on change]	

Specifies the system IP address

GATEWAY_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x007f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	read the gateway IP using string notation (xxx.xxx.xxx.xxx)	
Write	p_string	service (VJD only)	set gateway IP using string notation (xxx.xxx.xxx.xxx) [automatic application restart on change]	

Specifies the system gateway IP address.

SUBNET_STR

	Tag code	NumDes	Message	SNMP Support
	0x007d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	get current subnet mask string notation (xxx.xxx.xxx.xxx)	
Write	p_string	service (VJD only)	set subnetmask string notation (xxx.xxx.xxx.xxx) [automatic application restart on change]	

Specifies current system subnet mask.

DEVICE_IP_ADDRESS

	Tag code	NumDes	Message	SNMP Support
	0xd052	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	get network configuration	
Write	p_octet	service (VJD only)	set current network configuration [automatic application restart on change]	

Get/set network configuration.

Payload:

6 byte MAC address, 2 bytes reserved

4 bytes IPv4 address

4 bytes IPv4 subnet mask

4 bytes IPv4 gateway

128 bytes reserved

1 byte selector, 3 bytes reserved

Used selector bits for write commands (all other fields will be ignored in set requests):

Bit0 (0x01): Set IPv4 address

Bit1 (0x02): Set IPv4 subnet mask

Bit2 (0x04): Set IPv4 gateway

Note: Use selector 0x07 to configure all network settings in one step.

DHCP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x00af	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the DHCP state (0=OFF, 1=ON)	
Write	t_octet	service (VJD only)	set DHCP ip configuration (0=OFF, 1=ON) [automatic application restart on change]	

DNS_SERVER_IP_STRING

	Tag code	NumDes	Message	SNMP Support
	0x0b49	yes	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	get DNS server IPv4	
Write	p_string	service (VJD only)	set DNS server IPv4	

Num Descriptor Values

Primary DNS	0 or 1
Secondary DNS	2

DNS server settings are applied asynchronously. DNS server IPv4 read requests may return empty or older values after changing the settings.

HOST_NAME

	Tag code	NumDes	Message	SNMP Support
	0x0cda	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	noprot	read the hostname	
Write	p_unicode	service (VJD only)	set host name (max 128 unicode characters) [automatic application restart on change]	

Hostname can only be changed in extended configuration mode. Device will reboot on hostname changes.

LOCAL_HTTP_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0954	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	read http control port number	
Write	t_word	service	set http control port number [automatic application restart on change]	

0: Disable http control port 80

80: Enable http control port 80 (default)

LOCAL_HTTPS_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0a0e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	read https control port number	
Write	t_word	service	set https control port number [automatic application restart on change]	

0: Disable https control port 443

443: Enable https control port 443 (default)

RCP_SERVER_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0a17	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	read rcp+ control port number	
Write	t_word	service	set rcp+ control port number [automatic application restart on change]	

0: Disable rcp+ control port 1756

1756: Enable rcp+ control port 1756 (default)

MIN_TLS_VERSION

	Tag code	NumDes	Message	SNMP Support
	0x0c56	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	read minimum required TLS version	
Write	t_word	service	set minimum required TLS version [automatic application restart on change]	

0x0100: TLS 1.0

0x0102: TLS 1.2 (default)

RTSP_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0a63	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	noprot (VJD only)	read rtsp port number	
Write	t_dword	service (VJD only)	set rtsp port number [automatic application restart on change]	

0: Disable rtsp port 554

554: Enable rtsp port 554 (default)

RTSPS_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0ce6	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	noprot (VJD only)	read rtsps port number	
Write	t_dword	service (VJD only)	set rtsps port number [automatic application restart on change]	

0: Disable rtsps port 9554

9554: Enable rtsps port 9554 (default)

Time

DATE_WDAY

	Tag code	NumDes	Message	SNMP Support
	0x0027	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	"Sunday" ... "Saturday"; read the weekday according to the systems date setting	
Write			not supported	

Gets the current system weekday.
For example, if date is 01/17/2013, result will be “Thursday”.

DATE_DAY

	Tag code	NumDes	Message	SNMP Support
	0x0028	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the day of month	
Write			not supported	

Gets the current system day.
Example: Date is 1/17/2013. Result is 17

DATE_MONTH

	Tag code	NumDes	Message	SNMP Support
	0x0029	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the month	
Write			not supported	

Gets the current system month.
Example: Date is 1/17/2013. Result is 1

DATE_YEAR

	Tag code	NumDes	Message	SNMP Support
	0x002a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	read the year	
Write			not supported	

Gets the current system year;
Example: Date is 1/17/2013. Result is 2013

TIME_HRS

	Tag code	NumDes	Message	SNMP Support
	0x002d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the hours	
Write			not supported	

Gets the current system count of hours;
Example: Time is 14:35:48. Result is 14

TIME_MIN

	Tag code	NumDes	Message	SNMP Support
	0x002c	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the minutes	
Write			not supported	

Gets the current system count of minutes.
Example: Time is 14:35:48. Result is 35

TIME_SEC

	Tag code	NumDes	Message	SNMP Support
	0x002b	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	noprot	read the seconds	
Write			not supported	

Gets the current system count of seconds.
Example: Time is 14:35:48. Result is 48

TIMEZONE

	Tag code	NumDes	Message	SNMP Support
	0x024e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	noprot	the timezone in which the unit has to operate (UTC +-nbr of seconds +- nbr of seconds DLS)	
Write			not supported	

Gets the current system time zone as difference in seconds from UTC (including daylight saving time adjustment).
For example, UTC+02:00 is represented as 7200; UTC-01:00 is represented as -3600

TIME_ZONE_STRING

	Tag code	NumDes	Message	SNMP Support
	0xd027	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	noprot	on read direction the command returns all possible localized timezone strings, delimited by ";". The first timezone string denotes the current timezone and is therefore contained twice in the payload.	
Write	p_unicode	service (VJD only)	on write direction the command sets the timezone determined by the timezone string in the payload.	

NTP_SERVER_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x024f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	get the NTP server URL	
Write	p_string	service (VJD only)	set the NTP server URL	

The NTP server URL can be a plain IPv4 address, the NTP server port is not part of the URL (always 123). If provided, the NTP server URL will be automatically clipped to remove the port ":123".

FORCE_TIME_SET

	Tag code	NumDes	Message	SNMP Support
	0x0a0f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	noprot	read the time, 8 bytes payload, offset 0: year (word); offset 2: month (octet); offset 3: day (octet); offset 4: hrs (octet); offset 5: min (octet); offset 6: sec (octet); offset 7: reserved (octet)	
Write	p_octet	service (VJD only)	set the time, parameter 8 bytes payload, offset 0: year (word); offset 2: month (octet); offset 3: day (octet); offset 4: hrs (octet); offset 5: min (octet); offset 6: sec (octet); offset 7: reserved (octet) [may lead to an automatic application restart on change]	

SYSTEM_DATETIME_V2

	Tag code	NumDes	Message	SNMP Support
	0x0ba8	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	noprot	Gets the UTC system date and time: yyyy-mm-ddThh:mm:ss;; Timezone can only be retrieved via TIME_ZONE_STRING	
Write	p_string	service (VJD only)	Writes the UTC system date and time: yyyy-mm-ddThh:mm:ss;; [may lead to an automatic application restart on change] Timezone can only be changed via TIME_ZONE_STRING	

Connection

PASSWORD_SETTINGS

	Tag code	NumDes	Message	SNMP Support
	0x028b	password level	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	get the unit password (scrambled). num parameter sets the password levels; 1=user, 2=service	
Write	p_string	service	set the unit password. num parameter sets the password levels; 1=user, 2=service. Max length is 127 characters.	

Configures password for specified access level.

Type: Read command returns obfuscated password.

There are two different password levels:

- user (1)
- service (2)

ALARM_CONNECT_TO_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x0081	password table index	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	read IP using string notation (xxx.xxx.xxx.xxx)	
Write	p_string	service	set IP using string notation (xxx.xxx.xxx.xxx)	

Specifies the password table index for IP addresses of password-protected devices that connect to the Monitor Wall. Supported password index table number are 1 to 9.

The corresponding passwords need to be added by command REMOTE_PASSWORD.

REMOTE_PASSWORD

	Tag code	NumDes	Message	SNMP Support
	0x010c	password table index	no	no
	Datatype	**Access Level**	**Description**	
Read	void	user	obfuscated password	
Write	p_string	service	deposit the password of the called station. Max length is 19 characters	

Specifies the passwords of password-protected devices that the Monitor Wall shall be able to connect to. For Bosch/IQSIGHT cameras, Bosch/IQSIGHT encoders and Bosch/IQSIGHT network services, the “user” account passwords needs to be configured (least privileges policy).

Read command returns an obfuscated password.

Password table indexes 1 to 9 are reserved for IP addresses added via ALARM_CONNECT_TO_IP_STR, the password table entry at index 10 is used for all those IP addresses that have not been configured via ALARM_CONNECT_TO_IP_STR.

NBR_OF_ALTERNATIVE_ALARM_IPS

	Tag code	NumDes	Message	SNMP Support
	0x0303	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	noprot	get the number of available alarm ip addresses (total presets)	
Write	void	service	not supported	

Returns the number of supported passwords for devices (here: always 10)

ALARM_CONNECT_TO_IP

	Tag code	NumDes	Message	SNMP Support
	0x0041	destination IP number	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user	specifies the IP address	
Write	t_dword	service	specifies the IP address	

Specifies the password table index for IP addresses of password-protected devices that connect to the Monitor Wall. Supported password index table number are 1 to 9.
For password-protected Bosch/IQSIGHT /IQSIGHT cameras, Bosch/IQSIGHT encoders and Bosch/IQSIGHT network services, the “user” account will be used for login (least required privileges policy), thus the according “user” account passwords need to be added via REMOTE_PASSWORD.

CONNECT_TO

	Tag code	NumDes	Message	SNMP Support
	0xffcc	no	yes	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	see detailed description	

Connects media streams from a device. For password-protected Bosch/IQSIGHT cameras, Bosch/IQSIGHT encoders and Bosch/IQSIGHT network services, the “user” account will be used for login (least required privileges policy), thus the according “user” account passwords need to be added via REMOTE_PASSWORD.

Write Packet

Destination IP Address 4 Bytes		
Reserved 1 Byte	**Reserved** 1 Byte	**Flags** 2 Bytes
Local Coder 1 Byte	**Local Line** 1 Byte	**Put Channels** 2 Bytes
Remote Coder 1 Byte	**Remote Line** 1 Byte	**Get Channels** 2 Bytes

Destination IP Address

The camera's or encoder's IP address.

Flags

Values:

Bit0	Reserved
Bit3	Force the use of TCP as transportation protocol
Bit6	Connect a VCA meta data stream in addition to video
Bit8	Use SSL for the RCP control connection; if no destination port is specified, the camera's port defaults to 443 (HTTPS)
Bit9	Request video mode H.264
Bit10	Request video mode JPEG
Bit13	Request video mode H.265

NOTE: Only one video mode can be used; setting all bits will result in best currently available mode.

Local Coder

This parameter carries the decoder's one-based coder index (tile index).

Local Line

This parameter carries the decoder's one-based video output line index (screen index).

Put Channels

Not supported

Remote Coder

This parameter carries the camera's encoder index. Use 0 as a wildcard for best match.

Remote Line

This parameter carries the camera's one-based line index.

Get Channels

Values:

Bit0	Video. Ignored in this command, always active.
Bit1	Audio
Bit2	Data. Ignored in this command.
Bit3-Bit15	Reserved

NOTE: Audio connections are only supported on screen 1, decoder 1.

Optional Appendix

Destination Port 2 Bytes	**Reserved** 2 Bytes
Reserved 4 Bytes	

Destination Port

This parameter carries the camera's port number for the RCP login.

Reply / Message

The reply to this command will have the same content as the request.

A message will be generated if all requested channels are established; if a channel fails, the appropriate bit will be cleared in the channel section.

DISCONNECT_PRIMITIVE

	Tag code	NumDes	Message	SNMP Support
	0xff0d	no	yes	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet, f_flag	user	see detailed description	

Disconnecting video streams from the application

Reply Payload Structure

Status 1 Byte	Cause 1 Byte	Reserved 2 Bytes
Remote Host IP 4 Bytes		

Status

Values:

Connection disconnected	0x01
Connection identified by the given Session ID not found on this host	0x02

Cause

Values:

Not closed	0x00
Normal termination	0x01
Abnormal termination	0x02
No response	0x03
Remote host terminated	0x04
Timed out	0x05
Remote login rejected	0x06
No common media channels	0x07
Connection substituted	0x08
Automatic disconnect	0x09
Stop streaming	0x0a

Remote Host IP

IP address of the remote connected host.

NOTE: This command is NOT readable.

ACTIVE_CONNECTION_LIST

	Tag code	NumDes	Message	SNMP Support
	0xffc1	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	see detailed description	
Write			not supported	

Gets list of all active connections.

Payload Structure

Sequence of:

Destination IP Address 4 Bytes		
Local Coder 1 Byte	**Local Line** 1 Byte	**Flags** 2 Bytes
Session ID 4 Bytes		
Remote Coder 1 Byte	**Remote Line** 1 Byte	**Destination Port** 2 Bytes
TX Channels 4 Bytes		
RX Channels 4 Bytes		

Destination IP Address

IP address to which the unit is connected.

Local Coder

The coder number where the stream is connected to.

Local Line

The line number where the stream is connected to.

Flags

Values:

Bit8	Connection is H.264
Bit9	Connection is Jpeg
Bit11	Connection is H.265

Session ID

Session identifier.

Remote Coder

The connected camera encoder number (relative to the camera line).

Remote Line

The connected camera line number.

Destination Port

Contains the optional. Default is zero.

TX Channels

See CONNECT_TO command for bit mask.

RX Channels

See CONNECT_TO command for bit mask.

CONNECT_URL

	Tag code	NumDes	Message	SNMP Support
	0xe20c	read request only	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	URL with optional parameters	
Write	p_string	user	URL with optional parameters	

Connect a video panel to a device with the given url. The url may contain a password for write commands. If the default password shall be inserted automatically, use "*" as password in the url (without quotation marks). Digital zoom settings (centerX, centerY, scale) and dewarping settings (pan, tilt, scale) in response payloads reflect the current settings, which may have changed e.g. due to PTZ keyboard user actions (see PTZIF write command).

Numerical descriptor (read request only): one-based screen index in upper 8 bit, one-based tile index in lower 8 bit

String format (separator: space character):
<url> [<option1>=<value1>] ... [<optionN=valueN>]

URL format:
`<rcpp|http|https|rtsp|rtsps>://[<username>[:<password>]@]<ipAddress>[:<port>][/<path>]`

Default ports:

- rcpp 1756
- http 80
- https 443
- rtsp 554
- rtsps 9554

Optional parameters:

- `screen=<1..>` Monitor Wall line (default when omitted: 1)
- `tile=<1..>` Monitor Wall coder (default when omitted: 1)
- `type=<VIP|RTSP|ONVIF>` Device proxy type, `VIP` is the default when omitted
- `line=<1..>` Camera line number (default when omitted: 1)
- Either
  - `coder=<1..>` Camera encoder number (default when omitted: 1). Not supported for some urls.
  - `protocol=<tcp|udp>` Streaming protocol, see FALLBACK_STREAMING_PROTOCOL when omitted
  - `multicast=<true|false>` Multicast flag (default when omitted: true for rcpp urls, else false)

  or (only in conjunction with `type=ONVIF`)
  - `onvifProfileToken=<…>` ONVIF profile token string
  - `onvifTransportProtocol=<RTSP|TCP|UDP|RtspOverHttp|RtspUnicast|RtspMulticast>` ONVIF transport protocol, see FALLBACK_STREAMING_PROTOCOL when omitted
- `audio=<uint32>` Audio flags (0x0: no audio streaming (default when omitted); 0x1: connect audio stream, only in combination with `screen=1` and `tile=1`)
- `preset=<int32>` Dome preset to be applied after successful connection
- `centerX=<float 0..1> centerY=<float 0..1> scale=<float>` Digital zoom view parameters
- `pan=<float> tilt=<float> scale=<float>` Dewarping zoom view parameters

Examples:

```
rcpp://user:*@192.168.1.2
https://user:*@192.168.1.2:443 screen=1 tile=2 line=1 type=VIP
```

VERIFY_URL

	Tag code	NumDes	Message	SNMP Support
	0xe25b	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	URL with optional parameters	
Write	p_string	user	URL with optional parameters	

Verifies the connectivity to a device with the given URL, including authentication, streaming and decoding. When called by a client, a VERIFY_URL write command connects a camera analogue to CONNECT_URL. The same optional parameters can be added to the URL (details see CONNECT_URL), beside the screen and tile parameters. The connection will be established on screen 1 and tile 1, any existing connection on this panel will be closed and replaced. The write command returns the string “Ok” when the provided URL and options were meaningful to the Monitor Wall application. If not, the write command will return as payload an error description in English.

Establishing control and streaming connections is an asynchronous process, i.e. the write command cannot wait for a successful connection before returning to the caller. In case that the write command returns “Ok”, the client application shall repeatedly post VERIFY_URL read requests, e.g. every 100 milliseconds, until a non-empty string is returned. It is guaranteed that one read request will return with a non-empty string after a write request has returned “Ok”. Once the control and streaming connections were established successfully, the read request returns “Ok” after the first decoded video frame was rendered on screen 1, tile 1, which indicates that the connectivity has been verified successfully. With returning “Ok”, the connection to the given URL will be terminated automatically, so the video will only be displayed for a few milliseconds. In case that something went wrong during the asynchronous connection setup, authentication, decoding, etc., the read request will return as payload an error description in English.

FALLBACK_STREAMING_PROTOCOL

	Tag code	NumDes	Message	SNMP Support
	0xe25f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	set value	
Write	t_int	service	read value	

Defines the video streaming protocol preference for any video stream connect requests that omit the desired streaming protocol. Default value is 0 (use default fallbacks).

Values:

0 Default fallbacks (i.e. TCP for http/https URLs, and UDP/multicast for rcpp URLs)

1 Fallback to UDP/multicast

2 Fallback to UDP/Unicast

3 Fallback to TCP

VIDEO_OUTPUT_STATUS

	Tag code	NumDes	Message	SNMP Support
	0xe20e	no	yes	no
	Datatype	**Access Level**	**Description**	
Read	t_word	noprot	not supported, message only	
Write			not supported	

Signals changes of the video output

Payload Structure

Local Coder	Local Line	Status
1 Byte	1 Byte	2 Bytes

Local Coder

The decoder's coder number.

Local Line

The decoder's line number.

Status

Values:

0x0001	Streaming connection established
0x0002	First frame decoded and displayed
0x1000	Connection failure

DISCOVERY_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0976	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	user	reading current value of port	
Write	t_word	service	set port for application discovery via multicast [automatic application restart on change]	

Specifies the discovery port.

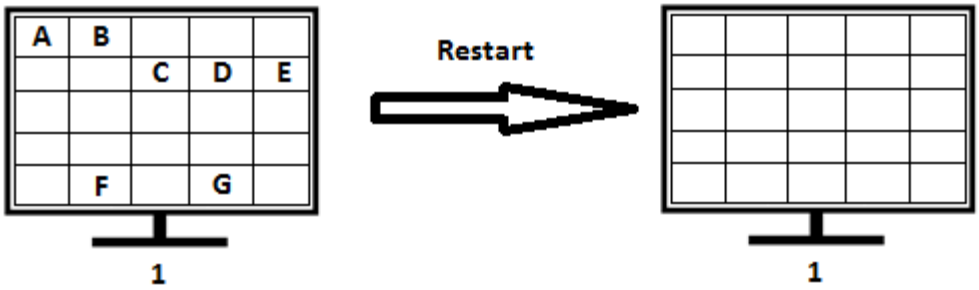
NOTE: Default value is 1800.

RECONNECT

	Tag code	NumDes	Message	SNMP Support
	0xe20a	No	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set reconnect 0=Disable; 1=Enable	

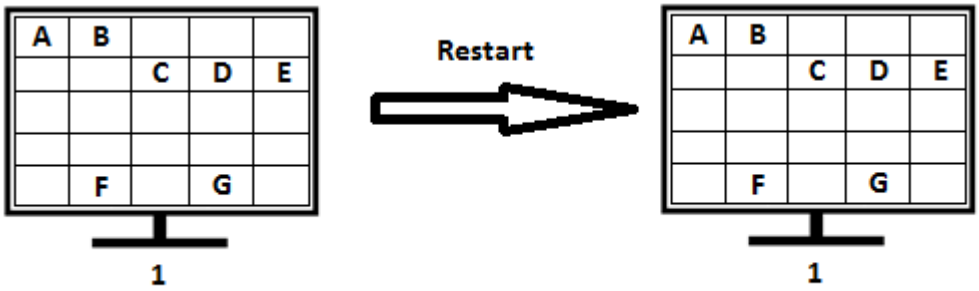
Specifies whether all streams should be reconnected after application restart.

If the parameter is 0, after restart all video streams will be lost:



A-G – connected video streams

If the parameter is 1, after restart all video streams will be reconnected at the same places:



Handling of digital and dewarping zoom settings at reconnect can be configured via DIGITAL_ZOOM_PERSISTENCE_MODE.

VIDEO_MEMORY_BUDGET

	Tag code	NumDes	Message	SNMP Support
	0xe259	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get video memory budget	
Write	t_int	service (MW only)	set video memory budget (only for workstations)	

The video memory budget reflects the memory limitations of a Videojet decoder device or workstation. Depending on the video resolution, each video stream is mapped to a certain video memory cost. Each newly connected video stream increases the video memory load on the decoder by its video memory cost. Please note that there is no linear relationship of the video memory budget, the video memory load and the video memory cost to the actual amount of memory that is required to keep the decoded video frames in the RAM.

The video memory load on the decoder must not exceed the video memory budget to avoid out-of-memory situations. This is enforced by the Monitor Wall application, which ignores further requests to connect video streams when the video memory budget will be exceeded after the request. Any such requests will return with an error. Please note that existing video streaming connections can still be replaced even when the video memory budget is fully used up.

The video memory budget for a decoder of the VJD-75x3 family is 132, with one exception: On a VJD-7533, connected to three or four high resolution displays (i.e. display resolutions beyond 1920x1440), the video memory budget drops to 48. The video memory budget of a workstation is by default not limited. In case that out-of-memory situations occur, the video memory budget may be configured to prevent future out-of-memory issues on the workstation.

The following table shows the mapping between video resolution and video memory cost:

Name	Video resolution range	Number of pixels per decoded video image	Video memory cost
SubHD	[0x0, 768x432]	[0, 331776]	3
HD	]768x432, 1280x720]	[331777, 921600]	4
FullHD	]1280x720, 1920x1080]	[921601, 2073600]	6
SubUHD	]1920x1080, 2992x1680]	[2073601, 5026560]	9
UHD	]2992x1680, 3840x2160]	[5026561, 8294400]	12

Please note that even when the video memory load stays within the video memory budget, it is not guaranteed that the decoder will not enter a computational overload state (I frame-only decoding state). For further details on the maximum number of “smooth” video streams in dependency of video codec, video resolution and video frame rate, please refer to the according VJD-75x3 datasheets.

Furthermore, replacing a video connection leads to a transient spike in the video memory load, since the new video connection is prepared in the background while the old video stream is still active. The spike ends when the existing video connection is teared down with the first-frame event of the new video connection. This transient overshooting of the video memory load is as designed.

Note: The available video memory budget may be reduced when a VOCS license is active on a Videojet decoder.

VIDEO_MEMORY_LOAD

	Tag code	NumDes	Message	SNMP Support
	0xe25a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current video memory load	
Write			not supported	

For further details, please refer to VIDEO_MEMORY_BUDGET.

VIDEO_MEMORY_PROTECTION_STATE

	Tag code	NumDes	Message	SNMP Support
	0xe260	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current video memory protection state	
Write	t_int	service	set current video memory protection state	

0: Video memory protection mechanism is disabled (default)

1: Video memory protection mechanism is enabled

For further details, please refer to VIDEO_MEMORY_BUDGET.

Monitor

BOARD_RESET

	Tag code	NumDes	Message	SNMP Support
	0x0811	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	void	service	restarts the application	

	Tag code	NumDes	Message	SNMP Support
	0x0811	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	t_octet	service	restarts the application or reboots the device	

Payload:

0x00 Restart application

0xFF Reboot device

DECODER_LAYOUT_LIST

	Tag code	NumDes	Message	SNMP Support
	0x09a1	line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	list of supported layouts. Each element has 2 bytes	
Write			not supported	

Gets the layout list for specified line.

The layouts are defined via a 16 bit layout descriptor. This descriptor can be used to set the current layout with DECODER_LAYOUT.

The scheme defined below describes in detail the Monitor Wall layouts based on square arrays with same number of columns and rows. This base concept is extended by a set of smaller video panels that always occupy a vertical and a horizontal edge (thumbnail border) in order to keep the overall aspect ratio.

For monitors which aspect ratio is deviating from the configured video panel aspect ratio, extra columns are inserted to the right. It is not possible to control this insertion via the API.

The concept defines a main area containing a monitor array plus an optional thumbnail border pair. A monitor layout is specified via a 16 bit value that is assembled from four times four bit values

15 12	11 8	7 4	3 0
extra columns	orientation	thumbnails	array

The table below describes the meaning of sixteen possible values of the 4 four-bit codes.

value	Thumbnail border orientation	thumbnail count	array	extra columns
0	top and left	no thumbnails		None
1	left and bottom		single	1
2	bottom and right		2x2	2
3	right and top		3x3	3
4			4x4	4
5		5	5x5	5
6			6x6	5
7		7	7x7	7
8			8x8	6
9		9		9
10				10
11		11		11
12				12
13		13		13
14				14
15		15		15

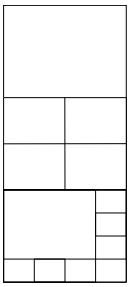
Supported layouts depend on physical characteristics of monitor and on the set aspect ratio.
For example, for a monitor with resolution 1920x1080 (display aspect ratio 16:9), the following layouts are supported per configured video panel aspect ratio. Other monitor resolutions may have different values. Any monitor resolutions are supported.

Aspect ratio	Layouts on a monitor with aspect ratio 16:9										
16:9	1 x 1	2 x 2	3 x 3	4 x 4	5 x 5	1 + 5	1 + 7	1 + 9	1 + 11	1 + 13	1 + 15
4:3	1 x 1	2 x 2	4 x 3	5 x 4	6 x 5	1 + 8	1 + 11	1 + 14	1 + 23	1 + 27	1 + 31
9:16	3 x 1	6 x 2	9 x 3	12 x 4	15 x 5	1 + 23	1 + 39	1 + 59	1 + 83	1 + 118	1 + 152
3:4	2 x 1	4 x 2	7 x 3	9 x 4	11 x 5	1 + 17	1 + 27	1 + 39	1 + 59	1 + 76	1 + 95

The corresponding layout codes are:

Aspect ratio	Layout’s codes on a monitor with aspect ratio 16:9								
16:9	0x0001	0x0002	0x0003	0x0004	0x0005	0x0251	0x0271	0x0291	0x02b1
	0x02d1	0x02f1							
4:3	0x0001	0x0002	0x1003	0x1004	0x1005	0x1251	0x1271	0x1291	0x22b1
	0x22d1	0x22f1							
9:16	0x2001	0x4002	0x6003	0x8004	0xa005	0x6251	0x8271	0xa291	0xc2b1
	0xf2d1	0x112f1f							
3:4	0x1001	0x2002	0x4003	0x5004	0x6005	0x4251	0x5271	0x6291	0x82b1
	0x92d1	0xa2f1							

Examples:



ID: 0x0001

ID: 0x0002

ID: 0x0271

DECODER_LAYOUT

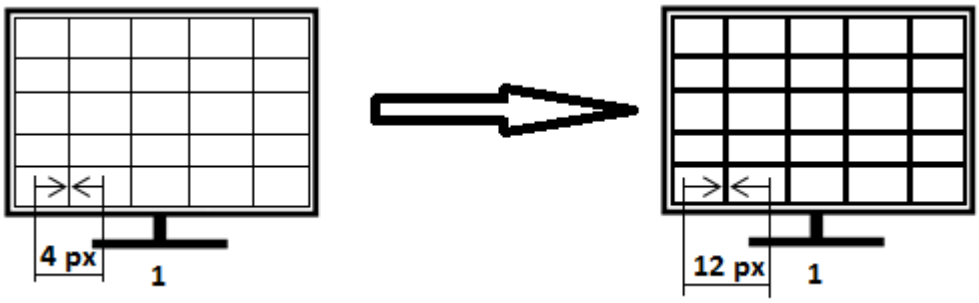
	Tag code	NumDes	Message	SNMP Support
	0x09a2	line	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	first 2 bytes are the layout, followed by the coder list in bytes (000101 for first monitor single-view, 000201020304 for quad-view with chronological order)	
Write	p_octet	user	first 2 bytes are the layout to be set, all further bytes will be ignored	

Gets the current layout for specified line.
Command uses the same layout codes as DECODER_LAYOUT_LIST command

CAMEO_DISTANCE

	Tag code	NumDes	Message	SNMP Support
	0xe209	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set border size of video panels (cameos)	

Specifies the video panel border size. Default value is 4px, this implies that the video panel spacing is 8px on the screen.
For example, parameter was changed from 2px to 6px:



NOTE: With a border size of 0px the blinking red alarm borders won't be rendered anymore.

CAMEO_ALARM_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe22c	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set alarm mode of video panels (cameos)	

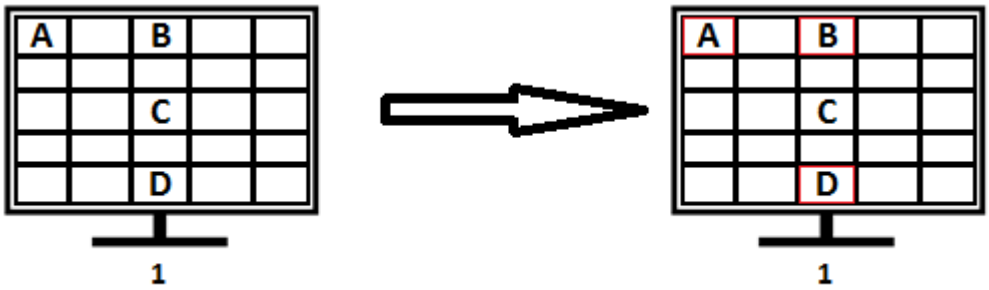
Specifies the video panel border behavior in case of a camera alarm event. Camera alarm events are either a signal loss event, a motion detection event, or an alarm event. Virtual alarm triggers are supported via the CAMEO_ALARM command.

0: Alarms are suppressed

1: [Default] Alarms are displayed as blinking custom-colored borders around the respective video panel for some seconds

2: Alarms are displayed as static custom-colored borders around the respective video panel for some seconds

Example of custom-colored borders (default color is red):



A-D – connected video streams.

A, B, D – video streams with motion alarms.

Note: Video panel alarm borders will not be rendered when the CAMEO_DISTANCE is too small.

CAMEO_ALARM_FRAME_COLOR

	Tag code	NumDes	Message	SNMP Support
	0xe267	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user	get current value	
Write	t_dword	service	set alarm frame color of video panels (cameos)	

Specifies the video panel alarm frame color as 32bit RGB color value (0x00RRGGBB, alpha channel is not supported). Default value is red (0x00FF0000).

CAMEO_ALARM

	Tag code	NumDes	Message	SNMP Support
	0xe268	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	triggers an alarm for a specific video panel with a certain duration	

Payload Structure

Line	Coder	Duration in seconds
1 Byte	1 Byte	2 Bytes

SWAP_MONITOR

	Tag code	NumDes	Message	SNMP Support
	0xe207	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	get current screen order	
Write	p_octet	service	see detailed description [automatic application restart on change]	

Specifies the mapping between Windows desktops and logical line numbers. Desktops are provided by the Windows operating system for attached monitors, in this context one desktop is provided per connected monitor device. The Monitor Wall application enumerates the provided desktops with a one-based index, starting with the left-most desktop at index 1.

The payload consists of the one-based Windows desktop number for line 1 (1 byte), followed by the one-based Windows desktop number for line 2 (1 byte), and so on, for each supported line (independent of the number of actual active lines with connected monitor devices). All values in the payload must be unique (i.e. no repetitions in the desktop numbers).

Payload structure

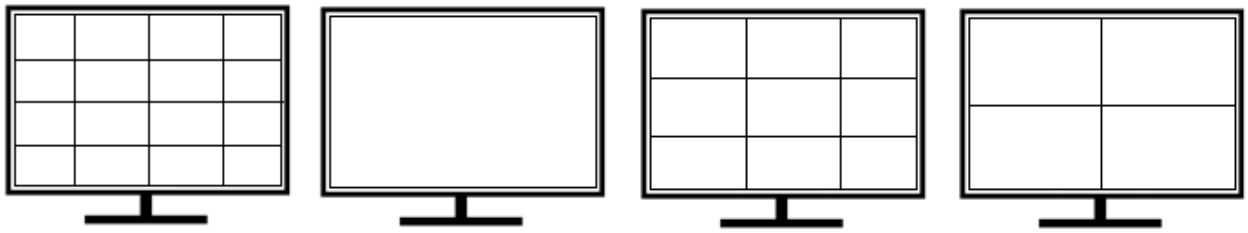
One-based Windows desktop number for line 1	... for line 2	...
1 Byte	1 Byte	

In order to achieve a certain desired order of Windows desktops (monitor devices) in a controlled way, the mapping between line number and Windows desktop number needs to be reset to its default setting as **first step**:

- For two supported lines (e.g. VJD-7513), send two bytes {0x01, 0x02} as payload
- For four supported lines (e.g. VJD-7523, VJD-7533), send four bytes {0x01, 0x02, 0x03, 0x04} as payload

With that, line 1 maps to Windows desktop 1 (left-most desktop), line 2 maps to Windows desktop 2 (next desktop to the right), and so on.

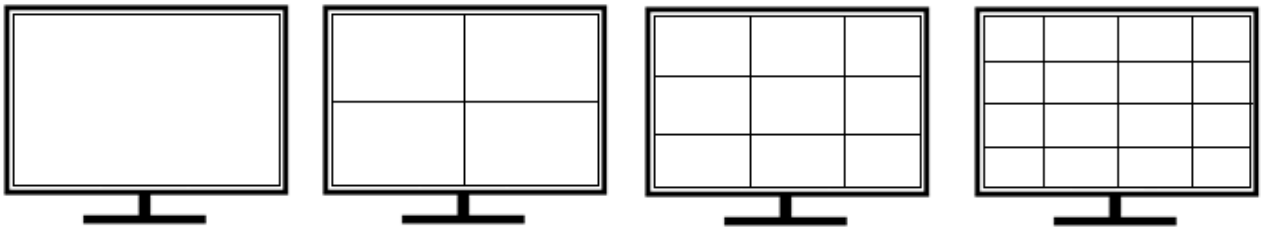
Then, as **second step**, the line number of each connected monitor needs to be identified, e.g. by assigning a certain video stream to a certain line, or by setting identifiable monitor layouts (e.g. use layout ID 1 on line 1, layout ID 2 on line 2, and so on, example see below), or by activating the monitor info overlay (either via PTZ keyboard or via INFO_OVERLAY_MODE).



The one-based Windows desktop numbers, from left to right for the monitor device arrangement in the example above, are given by: 4 (Line 4), 1 (Line 1), 3 (Line 3), 2 (Line 2)

Assuming that the desired monitor order shall begin with line 1 on the left-most monitor, and shall end with line 4 on the right-most monitor, the following payload needs to be sent in this example as **third step**: {0x04, 0x01, 0x03, 0x02}
The payload matches the identified order of Windows desktops from left to right.
Note: This command restarts Monitor Wall after returning the result of the write command.

After these three steps, the desired line arrangement is established:



Note: The ordering of monitors by Videojet decoders is based on the Windows mechanism for ordering desktops. In general, Microsoft Windows tries to reproduce the former desktop order after a reboot, with a focus on unique monitor device identifiers, typically independent of the display port or HDMI port where the monitor is plugged in. But this is not guaranteed when the connected monitor devices cannot be uniquely identified by EDID or serial number. Worst case scenario: The monitor devices do not support unique identifiers in the EDID, all monitors have the same model and firmware version, and the monitors are not all powered up (or not connected) when the Videojet decoder is starting up. Then, the monitor ordering may change between different runtimes.

SHOW_METADATA

	Tag code	NumDes	Message	SNMP Support
	0xe201	no	no	no
	Datatype	**Access Level**	**Description**	
Read	f_flag	user	get current value	
Write	f_flag	service	enable/disable rendering of VCA overlay	

Specifies whether video content analysis metadata should be shown as video overlay.

Parameter value: 1 – Enable; otherwise – Disable.

SHOW_METADATA2

	Tag code	NumDes	Message	SNMP Support
	0xe26d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user	get current value	
Write	t_dword	service	configure rendering of VCA overlay	

Specifies how video content analysis metadata should be shown as video overlay. The payload is interpreted as independent flags. Activation of certain non-default VCA overlays in crowded scenes may affect video smoothness.

Flags:

0x00000001	Show detected object outlines
0x00000002	Show object bounding boxes
0x00000004	Show video analytics events
0x00000008	Show alarm states
0x00000010	Show object center trajectories
0x00000020	Show head detections
0x00000040	Show class info
0x00000080	Show object foot point trajectories
0x00100000	Show stationary objects
0x00200000	Show object velocities
0x00400000	Show object sizes
0x00800000	Show attributes
0x80000000	Show object IDs

Masks:

0x00000000	Off
0x00000085	Default

OVERLAY_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe243	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current overlay mode	
Write	t_int	service	set overlay mode [automatic application restart on change]	

The setting allows to configure the global overlay mode.
The following overlay modes are currently supported:

- 0 None
- 1 Text overlay (default, see chapter **Stamping**)
- 2 Banner overlay (VJD only, see chapter **Banners**)

INSIGHTS_OVERLAY_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe261	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set insights overlay mode [automatic application restart on change]	

Insights overlay provides detailed on-screen information for video sessions. Each overlay option can be enabled separately in the mode, the payload is interpreted as set of independent flags.
Flags:

0x0001	Render encoded and decoded frames cache overlay
0x0002	Render decoder type overlay
0x0004	Render frame info text overlay
0x0008	Render frame latency overlay
0x0010	Render decoding load balancing overlay

Masks:

0x0000	Insights overlay off
0x001F	Render all

AUTO_RESOLUTION_CHANGE

	Tag code	NumDes	Message	SNMP Support
	0xe204 / 0xe205	no	no	no
	Datatype	**Access Level**	**Description**	
Read	f_flag / t_octet	user	get current value	
Write	f_flag / t_octet	service (VJD only)	set automatic monitor resolution [automatic application restart on change]	

NOTE: The original datatype for 0xe204 is f_flag, but will be interpreted as t_octet (both datatypes have the size 1 byte). The new command 0xe205 has the correct datatype t_octet.

Specifies whether monitor resolution should be set automatically. The color depth is always 32bit.

0: Off

1: Highest resolution at 60Hz (default)

2: Highest resolution at 50Hz

3: Highest resolution at 30Hz (useful for 4K screens that do not support 60Hz)

4: Best resolution for 1080p60 video streams (smallest resolution at 60Hz with at least 1920x1080)

5: Best resolution for 1080p50 video streams (smallest resolution at 50Hz with at least 1920x1080)

6: Best resolution for 720p60 video streams (smallest resolution at 60Hz with at least 1280x720)

7: Best resolution for 720p50 video streams (smallest resolution at 50Hz with at least 1280x720)

8: Highest screen resolution which is best for 60fps video (30Hz or 60Hz screen frequency)

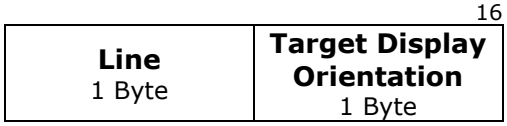
9: Highest screen resolution which is best for 50fps video (50Hz or 75Hz screen frequency)

TARGET_DISPLAY_ORIENTATION

	Tag code	NumDes	Message	SNMP Support
	0xe23d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	get current value(s)	
Write	p_octet	service (VJD only)	set target display orientation(s) [automatic application restart on change]	

Changing the display orientation will lead to an automatic restart of the Monitor Wall application. By default, the display orientation is not touched at startup (default target display orientation is "None"). Portrait modes are not supported by all monitor models, in such a case the Monitor Wall application nevertheless tries to set the according mode at each startup and fails, but without further consequence for normal operation (beside remaining in landscape mode).

The payload contains a sequence of two-byte-entries:



Read: Payload contains one two-byte-entry per supported line

Write: Add two-byte-entries for each line which display orientation shall be changed, i.e. the settings for multiple lines can be changed in parallel with a single write operation.

Supported display orientation modes:

0: None

1: Landscape

2: Portrait

3: Landscape (rotated)

4: Portrait (rotated)

SMOOTHNESS

	Tag code	NumDes	Message	SNMP Support
	0xe202	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set smoothness level for jittery video connections [0..100] [may lead to an automatic application restart on change]	

Specifies the global video de-jittering buffer configuration. A smoothness level of 0 minimizes the de-jittering buffer, thus resulting in minimum video latency but potentially non-fluent video due to network transmission issues or varying camera encoder workloads for different encoded frame types. A smoothness level of 100 allows automatic detection of the optimum de-jittering buffer length, thus resulting in smooth and fluent video but causing some delay (up to several 100ms even though optimized) until the video is displayed on the screen. Smoothness levels between 0 and 100 are compromises between the two extremes. Default setting is 100. Reducing this value may improve the perceived responsiveness of PTZ domes when using a joystick.

RENDERING_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe24d	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user (not for VJD-7513)	get current value	
Write	t_dword	service (not for VJD-7513)	set rendering mode [automatic application restart on change]	

Specifies the global rendering mode configuration:

0: Default (decoder selects upper limit for rendering frame rate according to datasheet)

1: Essential (upper limit for rendering frame rate is 15fps)

2: Fluent (upper limit for rendering frame rate is 30fps)

3: Best (upper limit for rendering frame rate is 60fps)

By default, the VJD-75x3 decoder is optimized to support the highest possible number of streams in parallel, even at the cost of reducing the rendering frame rate to 30fps or 15fps for some more stressful monitor configurations (see VJD-75x3 datasheet for more details). For some use cases, where only a small number of video streams is required, it may make sense to override the default configuration to achieve a more pleasing rendering frame rate of 30fps or even 60fps. Please be advised that the VJD-75x3 will then operate outside its specification, leading to a lower maximum number of supported streams in parallel than listed in the datasheet. The device cannot be damaged by adapting the rendering mode, but it may enter a processing overload and/or thermal overload state. These states can be detected with OVERLOAD_STATUS, and can be noticed in all displayed video streams, when decoding and rendering gets limited to I frames only to reduce the workload on the device.
The rendering mode can also be configured for workstations.

PREFERRED_ASPECT_RATIO

	Tag code	NumDes	Message	SNMP Support
	0xe203	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get current value	
Write	t_octet	service	set current aspect ratio. 0='4:3'; 1='16:9'; 2='9:16'; 3='3:4' [automatic application restart on change]	

This command is changing the aspect ratio of Monitor Wall video panels.

The standard values zero to three are defined as enumeration:

0='4:3'

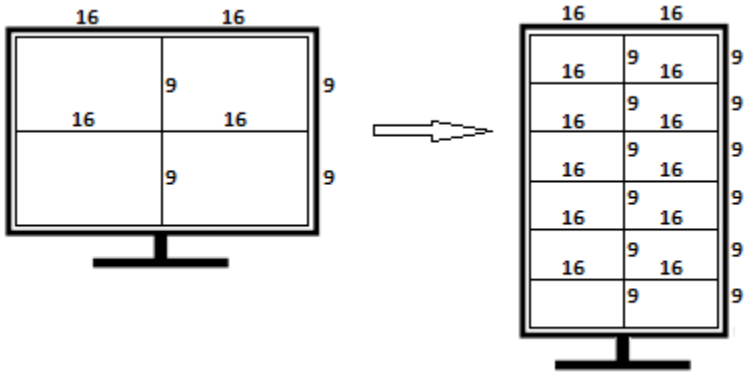
1='16:9'

2='9:16'

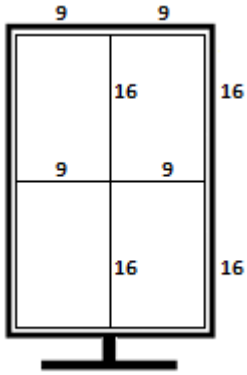
3='3:4'

The values above can also be achieved by configuring the preferred aspect ratio as width percentage of the height. So 4:3 can also be configured as 133, derived from 4/3 = 1.333… ≈ 133%. Any values between in the range [25, 400] are supported. Configuring the preferred aspect ratio to 100 would lead to square video panels.

For example, a video panel aspect ratio 16:9 is rendered on horizontal and vertical monitors as:



The aspect ratios 9:16 and 3:4 are primarily defined (but not exclusively) for rotated monitors as shown in the picture below:



NOTE: The Monitor Wall will perform a restart after changing the tile aspect ratio.

NUMBER_OF_DECODERS

	Tag code	NumDes	Message	SNMP Support
	0xe20b	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current limit	
Write	t_int	service	set current limit [automatic application restart on change]	

Specifies the maximum number of video panels across all lines that are supported by the device. The limit per line is obtained by dividing the value by the number of supported lines. The list of supported layouts may adapt to a changed number of decoders. Selection of layouts that exceed the number of decoders is still possible to certain degree, but video assignment to video panels beyond the configured number of decoders is not supported.

Default value for VJD-7513 is 30, for VJD-7523 and VJD-7533 it is 60, and for the Monitor Wall application it is 64. Maximum value for the VJD-75x3 is 99, and for the Monitor Wall application it is 128.

Changing the value has no effect on the current layout setting. Since clients may not expect the capabilities and decoder layout list to change during runtime, a restart of the client may be required.
NOTE: The Monitor Wall will perform a restart after changing the number of decoders.

ZOOM_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe20c	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set mode [automatic application restart on change]	

The following modes are currently identified:

0 Show all video content and keep aspect ratio (default)

1 Stretch the video to fill the video panel, the video aspect ratio may not be kept

DIGITAL_ZOOM_PERSISTENCE_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe23e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get current value	
Write	t_octet	service (VJD only)	set mode	

Three different types of pan-tilt-zoom (PTZ) modes are supported:

- Mechanical PTZ (dome cameras)
- Digital PTZ (fixed cameras)

- Dewarping PTZ (panoramic cameras)

Depending on the camera type, the PTZ mode is automatically preselected by the Monitor Wall application and cannot be overridden. Depending on the PTZ mode, the user input from the IP matrix PTZ keyboard joystick is translated to pan, tilt, and zoom speeds, which will then either control the dome camera PTZ motors, the rectangular region of interest of the video panel's digital zoom, or the video panel's dewarping zoom.
The digital zoom persistence mode defines what happens with digital and dewarping zoom states when a camera is connected again.

The following modes are supported:

0: Ignore

Previous digital and dewarping zoom settings will be ignored (default). Use this mode to avoid overrides e.g. when digital and dewarping zoom values are provided via CONNECT_URL.

1: Per source

Digital and dewarping zoom settings will be automatically stored per stream identifier (camera host name, camera video input line and camera encoder index (or ONVIF profile token)). Whenever a stream gets reconnected, the previous zoom settings will be restored, regardless of where or how often the stream gets displayed by the Videojet decoder.

2: Per source and sink

Digital and dewarping zoom settings will be automatically stored per stream identifier (camera host name, camera video input line and camera encoder index (or ONVIF profile token)) and per video panel identifier (line, coder). Whenever a stream gets reconnected on the same video panel, the previous zoom settings will be restored.

3: Per camera key

Digital and dewarping zoom settings will be automatically stored per camera key (see KBD_CONFIG_CAMERA_STR), but only when the video connection was established by camera key selection (see KBD_CONNECT_CAMERA). Whenever a video connection gets restored by camera key (see KBD_CONNECT_CAMERA and KBD_CONNECT_SALVO), the saved zoom settings will be restored.
Note: When a video connection is established by any other means (CONNECT_URL, reconnect at restart of Monitor Wall application, etc.), the zoom settings won't be restored.

Note: Changes to the zoom settings will not be persisted when the video connection is established during a camera salvo (see KBD_CONNECT_SALVO) or by any other means beside KBD_CONNECT_CAMERA.

Common to 1, 2, and 3: Per favorite key

Digital and dewarping zoom settings will be stored also per video panel identifier (line, coder) and per favorite key, but only when a favorite is stored (see SAVE_FAVORITE). Whenever a favorite gets restored (see LOAD_FAVORITE), the saved zoom settings will be restored.

Whenever the digital zoom persistence mode toggles, all stored zoom settings will be cleared, including all stored digital presets for all video sources (see KBD_STORE_PRESET).

IN_WIN_PTZ_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe25c	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service	set mode [automatic application restart on change]	

The Monitor Wall application always starts in full-screen mode, with a hidden mouse cursor by default. By changing the in-win-ptz mode, the mouse cursor can be made visible during Monitor Wall application runtime when the mouse is moved. When visible, the cursor icons indicate the direction of the PTZ control, per video panel. When pressing the left mouse button, PT (pan, tilt) action will occur in the direction as indicated by the mouse cursor icon, with a PT speed that is proportional to the distance between the mouse cursor and the video panel center. Z (zoom) action will occur when the mouse wheel position is changed while hovering over a video panel.
Depending on the type of connected video stream, the Monitor Wall application will decide whether to send PTZ control commands to the camera, to change the digital zoom setting of the video panel, or to change the client dewarping zoom settings (supported only for Bosch/IQSIGHT panoramic cameras). This feature can also be used in conjunction with the DIGITAL_ZOOM_PERSISTENCE_MODE feature.
Note: The Monitor Wall will restart when this mode is changed.

CONTEXT_MENU_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe26e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	user	get current value	
Write	t_dword	service	set mode [may lead to an automatic application restart on change]	

The Monitor Wall application always starts in full-screen mode, with a hidden mouse cursor by default. The mouse cursor can be made visible during Monitor Wall application runtime when the mouse is moved. The available context menu options can be configured with as separate and independent flags in the payload. When at least one of the flags for supported context menu options is set in the context menu mode, a context menu with standard behavior pops up on a mouse right button click.
Flags:

0x00000001	Camera info
0x00000002	Camera connections
0x00000004	Salvo connections
0x00000008	Monitor layout
0x00000010	Freezing
0x00000020	Fullscreen
0x00000040	Favorite loading
0x00000080	Favorite modification
0x00000100	Preset access
0x00000200	Preset modification

0x00000400	Connection modification
0x00010000	Screen swapping
0x00020000	Screen assignment mode
0x00040000	Aspect ratios
0x00080000	Screen settings
0x00100000	Mouse PTZ
0x00200000	Digital zoom persistence mode
0x00400000	Text overlay
0x00800000	Metadata overlay
0x01000000	Insights overlay
0x02000000	GPU decoding (MonitorWall only)
0x80000000	Context menu options

Masks:

0x00000000	Off
0x0000FFFF	Operations
0xFFFFFFFF	Operations and configurations

MONITOR_INFO

	Tag code	NumDes	Message	SNMP Support
	0xe20d	line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	see detailed description	
Write			not supported	

Get information about the line as a list of three four-byte DWORD values.

Info Structure

Width	Height	RefreshRate
4 Bytes	4 Bytes	4 Bytes

Width

Horizontal number of displayed pixels

Height

Vertical number of displayed pixels

RefreshRate

Display refresh rate in Hz

MONITOR_INFO2

	Tag code	NumDes	Message	SNMP Support
	0xe22d	line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	see detailed description	
Write			not supported	

Get information about the line as a list of five four-byte DWORD values.

Info Structure

Left	Top	Width	Height	RefreshRate
4 Bytes	4 Bytes	4 Bytes	4 Bytes	4 Bytes

Left

Horizontal position in display pixels

Top

vertical position in display pixels

Width

Horizontal number of displayed pixels

Height

Vertical number of displayed pixels

RefreshRate

Display refresh rate in Hz

MONITOR_POWER_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe20f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	user	get current value	
Write	t_int	service (VJD only)	set mode	

The setting allows to configure the desired monitor power save mode in case all video panels are unconnected.

This mode has no effect on the decoder device power state.

The following modes are currently defined:

- 0 Disabled (default). The monitor is powered on all time.
- 1 Sleep
- 2 Power Off

VOCS_CONFIG

	Tag code	NumDes	Message	SNMP Support
	0xe233	either line or 0 for all lines	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD-75x3 only)	get current config	
Write	p_octet	service (VJD-75x3 only)	set config [automatic application restart on change]	

The setting allows the configuration of the visual output capture service (VOCS) for one or more lines. The VOCS provides a H.264 encoded stream per line with original or reduced frame rate and size. For further details on the payload, please refer to VOCS_CONFIG2.

Numerical descriptor: 0
Payload format (6 Bytes per line):

Line	Frame rate divisor (log2)	Size divisor (log2)
2 Bytes	2 Bytes	2 Bytes

Numerical descriptor: Line
Payload format (4 Bytes):

Frame rate divisor (log2)	Size divisor (log2)
2 Bytes	2 Bytes

VOCS_CONFIG2

	Tag code	NumDes	Message	SNMP Support
	0xe257	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD-75x3 only)	get current config	
Write	p_octet	service (VJD-75x3 only)	set config [automatic application restart on change]	

The setting allows the configuration of the visual output capture service (VOCS) for one or more lines. The VOCS provides a H.264 encoded stream per line with original or reduced frame rate and size. The divisors are transmitted as logarithm with base 2 of the actual value, i.e. a transmitted value of 2 will result in a divisor value of 4 (i.e. log2(4) = 2). Certain flags can be set to affect the compression of the VOCS streams.

The VOCS stream target width (analogue: height) is derived by dividing the screen width with the actual size divisor. The actual size divisor is larger or equal to the configured size divisor, has the same range of values, and is chosen in a way that the maximum of target width and target height does not fall below 640 pixels.

The VOCS stream target frame rate is derived by dividing the screen refresh rate (e.g. 60Hz) with the rate divisor.

The target bitrate is chosen automatically, and is directly proportional to the target pixel rate (examples for default behavior: ~6Mbps for 1920x1080@30fps, ~750kbps for 960x540@15fps). The formula for the target bitrate can be affected with certain flags. The target bit rate is clipped to a minimum value of 200kbps.

Payload format (10 Bytes per line):

Line	**Frame rate divisor (log2)**	**Size divisor (log2)**	**Flags**	**Reserved**
2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes

Frame rate divisor: Log2 range [0,1,2,3,4] (-> divisor range [1,2,4,8,16])

Size divisor: Log2 range [0,1,2,3] (-> divisor range [1,2,4,8])

Flags:

0	Default
1	Half bit rate, lower VOCS video quality
2	Double bit rate, higher VOCS video quality

Reserved: 0

Snapshot download

Screen and tile snaphots can be downloaded via http GET /snap.jpg.

GET /snap.jpg[?JpegSize=<T-shirt size>]

Captures a screenshot for each connected display (screen), and stitches them into one overall bitmap if there is more than one display active. If only one screen is attached to the decoder, its screenshot will be taken directly as overall image. The overall image is then downscaled to the specified size (default size is S) and encoded as jpeg.

Example: https://<decoderIPv4>/snap.jpg?JpegSize=L

Example: https://<decoderIPv4>/snap.jpg

Supported snapshot sizes (T-shirt sizes):

XS	(extra small)	max width 240
S	(small)	max width 480, default setting
M	(medium)	max width 960
L	(large)	max width 1920
XL	(extra large)	original resolution

Note: The downscaling factor is a power of 2, i.e. the divisor for the width of the overall image is doubled iteratively as long as the downscaled width is larger than the selected max width value from the list of T-shirt sizes.

Note: The timeout for the whole snapshot GET request is 5s. The download is terminated by the server when the timeout is reached, which may happen for large snapshots in conjunction with low network bandwidths.

Note: The capture times may slightly differ for each attached display in a stitched image.

Note: The displayed video may stutter or flicker once during a snapshot request.

Further optional parameters may be provided for GET /snap.jpg (separated with a '&' character):

JpegScreen=<1..maxScreen>

JpegTile=<1..maxTile>

JpegQuality=<1..32>

Screen denotes the one-based line number, and tile denotes the one-based coder number. The return result depends on the number and the types of the provided optional parameters:

- JpegScreen only: A snapshot of the whole selected screen is returned, the original resolution is defined by the according display resolution.
  Example: https://<decoderIPv4>/snap.jpg?JpegScreen=2
- JpegScreen and JpegTile: A snapshot of the selected tile on the selected screen is returned, the original resolution is defined by the video stream's resolution.
  Example: https://<decoderIPv4>/snap.jpg?JpegScreen=2&JpegTile=3&JpegSize=XL
- JpegTile only: Captures a snapshot of the specified tile for each connected display (screen) with the connected video stream's resolution (with slight variations in the capture times), and stitches them horizontally into one overall bitmap (starting with screen 1 at the left) if there is more than one

display active. If only one screen is attached to the decoder, the tile snapshot of screen 1 will be taken directly as overall image.

Example: https://<decoderIPv4>/snap.jpg?JpegTile=2

The quality ranges from 1 (best) to 32 (modest). If this parameter is omitted, snapshots of good quality will be generated.

Note: Whenever a snapshot is not possible (the specified monitor is not attached, the specified tile is located outside of the active tile layout, no video source connected to the specified tile, etc.), a gray image with original size 640x480 will be used as placeholder.

Background logo up- and download

Static background logos for (A) unconnected tiles and (B) tiles with video connection issues can be uploaded via (A) http PUT /Logo.jpg and (B) http PUT /NoCamLogo.jpg, download is supported via (A) http GET /Logo.jpg and (B) http GET /NoCamLogo.jpg.
Dynamic background logos (animated gifs) for (A) unconnected tiles can be uploaded via http PUT /Logo.gif, download is supported via (A) http GET /Logo.gif.
The logo is then displayed as background for each unconnected video panel, and the no-cam logo gets displayed when a video connection issue occurs.
The logos can be cleared by either sending an image file with size 0kB, or by setting the logos that shall remain on the device via LOGO_LIST.

LOGO_LIST

	Tag code	NumDes	Message	SNMP Support
	0xe242	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	user	get current list of active logos	
Write	p_string	service	set list of logos that shall remain active [automatic application restart on change]	

The read command returns a semicolon-separated list of logos that have been uploaded. Depending on the uploaded logos, the payload is thus either empty or one of

- Logo
- Logo;NoCamLogo
- NoCamLogo

The write command acts like a filter, any previously uploaded logo will be deleted if it is not found in the payload. For example, when sending an empty payload with the write command, both logos will be deleted.

Banners

On a Videojet decoder, banners are user-defined images that can be displayed on top of the video panels. Banners are always scaled so that the whole width of the screen is covered, and clipped symmetrically at top and bottom to keep the banner height less than one third of the screen height.
Up to 99 banners can be uploaded via http PUT, the only supported image format is jpeg. Each banner may have its own individual width and height. Available banners can be listed, removed and downloaded (download via http GET).
Uploaded banners can be combined into banner salvos with a common cycle time (dwell time) per banner salvo. The banner cycle time granularity is 1 second. A banner salvo can also include banner pauses, i.e. it is possible to not display a banner during one or more cycles in a banner salvo, e.g. to display a flashing banner.
Banners can be positioned at the top, in the mid, or at the bottom of the screen. The banner position can be selected per banner salvo, i.e. all banners in a salvo will be displayed at the same position. In total, up to three independent banner salvos are supported per screen (top, mid, bottom). Banner salvos are not synchronized across different positions. Whenever a banner salvo is applied to a position where another banner salvo is already active, the former banner salvo will be replaced.

Banner up- and download

Banners can be uploaded via http PUT /BannerXY.jpg, banner download is supported via http GET /BannerXY.jpg. The banner identifier range is 01 to 99, i.e. the leading zero is required so that the identifier has always two decimal digits.
Banners can be cleared by either sending an image file with size 0kB, or by setting the banners that shall remain on the device via BANNER_LIST.

BANNER_LIST

	Tag code	NumDes	Message	SNMP Support
	0xe241	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	get current list of uploaded banners	
Write	p_octet	service (VJD only)	set list of banners that shall remain available [automatic application restart on change]	

The read command returns a sequence of 2 byte banner identifiers (unsigned short, range [1..99]), and lists all identifiers of banners that have been uploaded.
The write command acts like a filter, any previously uploaded banner will be deleted if its identifier is not found in the payload (again a sequence of 2 byte banner identifiers). For example, when sending an empty payload with the write command, all banners will be deleted.

CONFIG_BANNER_SALVO

	Tag code	NumDes	Message	SNMP Support
	0xe23f	Banner salvo identifier [1-16]	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	read configuration of banner salvo	
Write	p_octet	service (VJD only)	configure banner salvo	

Payload Structure

Cycle Time 2 Bytes	Reserved 2 Bytes
Banner Identifier 1 2 Bytes	...

Cycle Time

Display duration (cycle time, dwell time) of each banner in seconds

Banner Identifiers

Sequence of up to 32 banner identifiers, banner identifier range [0..99].
NOTE: Banner identifier 0 indicates a banner pause, i.e. during that cycle no banner will be displayed

CONFIG_BANNER_SALVOS

	Tag code	NumDes	Message	SNMP Support
	0xe246	type filter	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	read whole banner salvo configuration	
Write	p_octet	service (VJD only)	configure banner salvos in a single step	

The banner salvo types can be selected by setting the corresponding bits in the numeric descriptor (type filter):

Bit0 (0x0001): Banner salvo with common cycle time

Read

The numeric descriptor (type filter) defines which banner salvo types shall be returned. E.g. 0x0001 will return only all banner salvos with a common cycle time. A type filer with value 0x0000 will be handled like 0xffff, i.e. all banner salvo types will be included.

Write

The numeric descriptor defines which banner salvo types shall be deleted from the config before the new banner salvos will be added. If a certain banner salvo type is not set in the numeric descriptor but an entry of that type is found in the payload, it will still be written into the config (either replacing an existing entry or added as new entry, all other entries remain unchanged).

The payload is a sequence of banner salvo configuration entries and their corresponding headers.

Payload Structure

Sequence of

Length 2 Bytes	**Type** 2 Bytes	**Banner Salvo Identifier** 2 Bytes	**Reserved** 2 Bytes	**Banner Salvo Configuration** Length-8 Bytes

Length

Total length of the banner salvo entry including the 8 byte header

Type

Banner salvo type (0x0001: Banner salvo with common cycle time)

Banner Salvo Identifier

Range [1-16]

Banner Salvo Configuration

{Length minus 8} bytes with the actual banner salvo configuration.
Banner salvo type 0x0001: See CONFIG_BANNER_SALVO for payload definition

CONNECT_BANNER_SALVO

	Tag code	NumDes	Message	SNMP Support
	0xe240	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	get overview of banner salvo(s)	
Write	p_octet	service (VJD only)	banner salvo(s) to be started or stopped	

Read Request Payload Structure

The read request payload can be either empty, to get banner salvo information on all lines and positions, or 4 bytes, to get information on a specific line and position:

Line 2 Bytes	**Position** 2 Bytes

Line

Video output line number

Position

Banner position. Supported positions are:

- 0 Top
- 1 Center
- 2 Bottom

Write and Read Response Payload Structure

Sequence of 6 byte entries

Line 2 Bytes	**Position** 2 Bytes	**Banner Salvo Identifier** 2 Bytes
...		

Banner Salvo Identifier

- 0 Write: Stop banner salvo at given line and position; Read: No banner salvo active at given line and position
- 1..16 Write: Start banner salvo with given identifier at given line and position; Read: Banner salvo identifier of active banner salvo at given line and position

Stamping

NAME_STAMP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x0084	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get current value	
Write	t_octet	service	set mode	

Defines the position of the overlay

- 0 stamping off
- 1 stamping on bottom
- 2 stamping on top
- 3 stamping with custom attributes

For custom position and other detail configuration see the corresponding ATTR command.

TIME_STAMP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x0085	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get current value	
Write	t_octet	service	set mode	

Defines the position of the overlay

- 0 stamping off
- 1 stamping on bottom
- 2 stamping on top
- 3 stamping with custom attributes

For custom position and other detail configuration see the corresponding ATTR command.

ALARM_DISP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x008e	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get current value	
Write	t_octet	service	set mode	

Defines the position of the overlay

- 0 stamping off
- 1 stamping on bottom
- 2 stamping on top
- 3 stamping with custom attributes

For custom position and other detail configuration see the corresponding ATTR command.

STAMP_ATTR_NAME

	Tag code	NumDes	Message	SNMP Support
	0x0936	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	set configuration	
Write	p_octet	service	get configuration	

x 1 Byte	**y** 1 Byte	**reserved** 2 Bytes
attributes 4 Byte		
reserved 2 Bytes		**flags** 2 Bytes
Color 4 Byte		
BackgroundColor 4 Byte		

x

Horizontal position relative to the Cameo (0= left, 255=right)

y

Vertical position relative to the Cameo (0= top, 255=bottom)

attributes

Values:

Bit 0 ... 11	reserved
Bit 12 ... 15	Height of the text in percent of the screen height (Range 15% - 1%)
Bit 16 ... 31	reserved

flags

Reserved for future use.

Color

32 Bit RGBA encoded color and transparency of the font.

BackgroundColor

32 Bit RGBA encoded color and transparency of the text background.

Note: the first 12 byte of this command is identical for cameras and encoders. Check the payload size before accessing the additional two 32bit color values.

STAMP_ATTR_TIME

	Tag code	NumDes	Message	SNMP Support
	0x0937	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	set configuration	
Write	p_octet	service	get configuration	

For the payload definition refer to STAMP_ATTR_NAME.

STAMP_ATTR_ALARM

	Tag code	NumDes	Message	SNMP Support
	0x0938	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	set configuration	
Write	p_octet	service	get configuration	

For the payload definition refer to STAMP_ATTR_NAME.

ALARM_STRING

	Tag code	NumDes	Message	SNMP Support
	0x0090	yes	no	yes
	Datatype	**Access Level**	**Description**	
Read	p_unicode	user	read the alarm string of a cameo	
Write	p_unicode	service	set an alarm string	

This command allows to set and get an alarm string of a video panel on a monitor.
The 16 bit numeric descriptor is used as follows:

Line	Coder
1 Byte	1 Byte

Keyboard and IP Matrix Configuration

KBD_CONFIG_CAMERA_STR

	Tag code	NumDes	Message	SNMP Support
	0x0ba3	camera table key [0-4095]	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read entry of camera table	
Write	p_octet	service	configure camera table entry	

Payload Structure

Line 1 Byte	Coder 1 Byte	Preset 1 Byte	URL Length 1 Byte
URL and Options N Bytes			

Line
Camera video input line

Coder
Encoder number

Preset (optional)
(Dome-) preset position. (Every preset of a dome can be treated as a separate camera).

URL Length
Length of the following url

URL
Encoder/Camera URL and space-separated options (subset of CONNECT_URL options, i.e. the screen and tile parameters are not supported)

NOTE: Options added to the url will override the values from the first three bytes of the payload.

KBD_CONFIG_MONITOR_STR

	Tag code	NumDes	Message	SNMP Support
	0x0ba4	monitor table key [0-4095]	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read monitor table entry	
Write	p_octet	service	configure monitor table entry	

Payload Structure

Line 1 Byte	Coder 1 Byte	URL Length 1 Byte	Reserved 1 Byte
URL N Bytes			

Line
Video output line number

Coder
Video panel (coder) number

URL Length
Length of the following url

URL
Length > 1: Decoder/Monitor URL

Length = 1 (VJD only): One byte with index of the decoder in the decoder group [0-3], see DECODER_GROUP

KBD_CONNECT_PARAMS

	Tag code	NumDes	Message	SNMP Support
	0x0a33	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read parameters for connections established via camera table	
Write	p_octet	service	set parameters for connections established via camera table	

Payload Structure

Flags 2 Bytes	Reserved 2 Bytes

Flags (optional)
Values:

Bit 0	Request audio TX (listen at decoder side)
Bit 1	Request audio RX (speak at decoder side)

KBD_PASSWORD_CAMERA

	Tag code	NumDes	Message	SNMP Support
	0x0a34	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	service	get the password (scrambled) for all cameras in the camera table	
Write	p_string	service	deposit a common password for all cameras in the camera table	

KBD_PASSWORD

	Tag code	NumDes	Message	SNMP Support
	0x0a69	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	service	get the password (scrambled) of the keyboard	
Write	p_string	service (VJD only)	deposit a password for the keyboard	

KBD_TIMEOUT

	Tag code	NumDes	Message	SNMP Support
	0x0a6a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int32		read keyboard lock timeout in minutes	
Write	t_int32	service (VJD only)	set keyboard lock timeout in minutes, default value is 10min, max value is 2880min (2 days), 0min disables lock timeout	

KBD_CONFIG_SALVO

	Tag code	NumDes	Message	SNMP Support
	0x0a3e	salvo table key [1-32]	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read configuration of salvo from salvo table	
Write	p_octet	service	configure salvo in salvo table	

Payload Structure

Duration 2 Bytes		**Line** 1 Byte	**Coder** 1 Byte
Camera Table Key 1 1 Byte	N x 1 Byte	**Camera Table Key N** 1 Byte	

Duration
Duration of one salvo position in seconds

Line
Video output line number

Coder
Video panel (coder) number

Camera Table Key N
Camera table key for salvo position N [0..255].
NOTE: Corresponding camera table entry has to be specified using KBD_CONFIG_CAMERA or KBD_CONFIG_CAMERA_STR

Salvo: A salvo is sequence of cameras that are cycled on one video panel (coder) with a fixed salvo cycle time (duration). The sequence is defined by a list of camera table keys. Salvos are stored in the salvo table.

Usage:
No longer adequate due to limited range of camera table key values. See KBD_CONFIG_SALVO2 for replacement.

KBD_CONFIG_SALVO2

	Tag code	NumDes	Message	SNMP Support
	0xe21e	salvo table key [1-32]	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read configuration of salvo from salvo table	
Write	p_octet	service	configure salvo in salvo table	

Payload Structure

<table>
<tr><td colspan="2">Duration
2 Bytes</td><td>Line
1 Byte</td><td>Coder
1 Byte</td></tr>
<tr><td>Camera Table Key 1
2 Bytes</td><td>N x 2 Bytes</td><td colspan="2">Camera Table Key N
2 Bytes</td></tr>
</table>

Duration
Duration of one salvo position in seconds

Line
Video output line number

Coder
Video panel (coder) number

Camera Table Key N
Camera table key for salvo position N [0..4095].
NOTE: Corresponding camera table entry has to be specified using KBD_CONFIG_CAMERA or KBD_CONFIG_CAMERA_STR

Salvo: A salvo is sequence of cameras that are cycled on one video panel (coder) with a fixed salvo cycle time (duration). The sequence is defined by a list of camera table keys. Salvos are stored in the salvo table.

Usage:

1) **Configure a salvo table entry for a given salvo table key:** Provide salvo table key as numeric descriptor, set duration, set line and coder to zero, append up to 256 camera table keys with size 2 bytes in the range [0..4095]

2) **Start a configured salvo on a selected video panel:** Select salvo table entry by setting the numeric descriptor to the salvo table key, set duration to zero (will be ignored), set line and coder to select target video panel, do not add any camera table keys
3) **Configure and start salvo in one step:** See 1), but set line and coder to select target video panel
4) **Stop a running salvo:** Set numeric descriptor to zero, set duration to zero (will be ignored), set line and coder to select the video panel which salvo shall be stopped, do not add any camera table keys
5) **Clear a specific salvo in the salvo table:** Provide salvo table key as numeric descriptor, set duration to zero (will be ignored), set line and coder to zero, do not add any camera table keys
6) **Clear all salvos in salvo table:** Set numeric descriptor to zero, leave payload empty

KBD_CONFIG

	Tag code	NumDes	Message	SNMP Support
	0xe21a	table entry type selection	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read configuration tables	
Write	p_octet	service	configure camera, monitor, and salvo tables	

The Monitor Wall configuration database contains tables for camera, monitor, and salvo descriptors. Each table entry can be referenced by a table key, e.g. by typing in the table key on a video control keyboard connected to the hardware decoder device. A camera table entry describes a network video source (url, video input line, etc), a monitor table entry describes a video panel (coder) on a screen (line) for a certain hardware decoder (url), and a salvo table entry defines a sequence of camera table keys used for automatic cycling through a list of cameras in one video panel.

The table entry types can be selected by setting the corresponding bits in the numeric descriptor:
Bit0 (0x0001): Camera
Bit1 (0x0002): Monitor
Bit2 (0x0004): Salvo

Read

The numeric descriptor defines which table entry types shall be returned. E.g. 0x0001 will return only all camera table entries, 0x0007 will return all camera, monitor, and salvo table entries (0x0000 will also return all entries).

Write

The numeric descriptor defines which table entry types shall be deleted from the config before the new table entries will be added. If a certain table entry type is not set in the numeric descriptor but an entry of that type is found in the payload, it will still be written into the config (either replacing an existing entry or added as new entry, all other entries remain unchanged). Setting the numeric descriptor to 0x0007 and leaving the payload empty will result in the deletion of all camera, monitor, and salvo table entries in the configuration database.

The payload is a sequence of camera, monitor, and/or salvo table entries and their corresponding headers.

Payload Structure

Sequence of

Length 2 Bytes	**Type** 2 Bytes	**Key** 2 Bytes	**Reserved** 2 Bytes	**Entry** Length-8 Bytes

Length
Total length of the table entry description including the 8 byte header

Type
Table entry type, 0x0001: Camera, 0x0002: Monitor, 0x0004: Salvo

Key
Table key, range [0-4095] for cameras and monitors, [1-32] for salvos

Entry
Table entry, given as sequence of bytes as defined in the playload sections of the commands KBD_CONFIG_CAMERA_STR, KBD_CONFIG_MONITOR_STR, and KBD_CONFIG_SALVO2 (usage 1).

DECODER_GROUP

	Tag code	NumDes	Message	SNMP Support
	0xe21d	no	no	No
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read decoder group configuration	
Write	p_octet	service (VJD only)	configure decoder group configuration	

Allows the configuration of a decoder group that will form a cluster of hardware decoders and physical screens that can be controlled with video control keyboards. A decoder group can consist of 1 to 4 decoders, each with 1 to 4 attached physical screens. The first decoder entry is the master of the decoder group, all following decoder entries are slaves.
Note: The IP matrix feature set requires static IP addresses for all decoders in a group. The read response will be empty if the static IP address is not configured, and the write will then return an error.
Please note that the decoders in the decoder group will synchronize the IP matrix configuration at decoder startup and during configuration.

Payload Structure

The payload is a sequence of 0 to 4 entries.
0 entries: Clears the decoder group configuration
1 to 4 entries, one entry per decoder in the decoder group:

- 4 bytes: hardware decoder IP address
- 1 byte: number of attached physical screens (range 1 to 2 for VJD-7513, range 1 to 4 for VJD-7523), defines also the number of non-zero 4 byte entries later in the payload
- 1 byte: number of supported lines (2 for VJD-7513, 4 for VJD-7523 and VJD-7533)
- 4 bytes per supported line (VJD-7513: 2 supported lines; VJD-7523, VJD-7533: 4 supported lines):
    - "non-zero 4 byte entry" if a physical monitor is attached to the according line:

- 2 bytes: monitor table start key (one-based, range [1..999])
- 2 bytes: maximum number of monitors [1..] on that screen

- "zero 4 byte entry" if no physical monitor is attached to the according line:
  - 4 bytes with 0x00

FREEZE_CHECK_INTERVAL

	Tag code	NumDes	Message	SNMP Support
	0xe220	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int32	user	read freeze check interval in seconds	
Write	t_int32	service	set freeze check interval in seconds, default value is 10s	

Each video panel is monitored separately for connection loss and frozen video. Whenever a connection loss is detected, the video connection is closed and will be established anew. Frozen video may recover, and can be interpreted as intermediate state before a connection loss is finally detected.
Connection loss will be displayed as grey background with the no-cam logo in the middle (video will be replaced), frozen video will be displayed as blinking red "FREEZE" text overlay on top of the last rendered video frame until the video resumes or a connection loss is finally detected.
It may happen that the default value of 10s for the freeze check interval is higher than the connection loss detection interval (which depends on the streaming protocol), then the freeze text overlay will not be displayed before the no-cam logo.

SALVO_PAUSE_TIMEOUT

	Tag code	NumDes	Message	SNMP Support
	0xe221	No	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int32	user	read salvo pause timeout in seconds (see SALVO_PAUSE_STATE)	
Write	t_int32	service	set salvo pause timeout in seconds, default value is 60s (see SALVO_PAUSE_STATE)	

KBD_CONFIG_CAMERA_MAPPING

	Tag code	NumDes	Message	SNMP Support
	0xe254	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	read mapping between internal camera indices and alias indices	
Write	p_octet	service	set mapping between internal camera indices and alias indices	

IP matrix configuration contains at least two lists of indices: one index list for a set of cameras and another index list that addresses monitors on the matrix. By using this command a user can replace the existing index list for cameras with a new list of camera indices.
Note: Alias indices must be provided for all cameras configured in the existing IP matrix.

Payload Structure

Payload is a sequence of pairs of unsigned shorts (2 Bytes), first ushort is the camera's existing index in the IP matrix, the second ushort is the camera's alias index.
There must be as many pairs as there are cameras configured in the IP matrix.

Camera index in IP matrix	**Alias camera index**
2 Bytes	2 Bytes

Keyboard and IP Matrix Operations

INFO_OVERLAY_MODE

Tag code	NumDes	Message	SNMP Support
0xe219	no	no	no

	Datatype	Access Level	Description
Read			not supported
Write	t_int32	user	info text overlay mode 0: off 1: video panel (coder) and screen (line) info 2: video source (camera) info 3: system load info

Info overlays time out and disappear after 10 seconds.

KBD_CONNECT_CAMERA

Tag code	NumDes	Message	SNMP Support
0xe211	yes	no	no

	Datatype	Access Level	Description
Read	p_octet	user	get camera key (0xFFFF: monitor is not connected to a camera from the camera table)
Write	p_octet	user	connect a camera from the camera table to a video panel (monitor)

Payload Structure

Monitor Key	Camera Key
2 Bytes	2 Bytes

Monitor Key

NumDes = 0: Monitor table key [0-4095]

NumDes >0: Line = NumDes, coder = monitor key

Camera Key

Camera table key [0-4095], can be omitted for read requests

KBD_CONNECT_SALVO

Tag code	NumDes	Message	SNMP Support
0xe212	yes	no	no

	Datatype	Access Level	Description
Read	p_octet	user	get salvo key (0xFFFF: monitor is not connected to a salvo from the salvo table)
Write	p_octet	user	connect a salvo from the salvo table to a video panel (monitor)

Payload Structure

Monitor Key 2 Bytes	Salvo Key 2 Bytes

Monitor Key
NumDes = 0: Monitor table key [0-4095]

NumDes >0: Line = NumDes, coder = monitor key

Salvo Key
Salvo table key [1-32]

KBD_GOTO_PRESET

Tag code	NumDes	Message	SNMP Support
0xe244	yes	no	no
Datatype	**Access Level**	**Description**	
Read		not supported	
Write	p_octet	user	go-to preset

Payload Structure

Monitor Key 2 Bytes	Preset 2 Bytes

Monitor Key
NumDes = 0: Monitor table key [0-4095]

NumDes >0: Line = NumDes, coder = monitor key

Preset
Number of preset to be activated

For PTZ domes, the preset number is sent to the device. If a preset has been stored beforehand with that number, the dome will restore the preset. If no preset is available or go-to preset is not authorized for the camera user, nothing will happen, no error code will be returned.

For all non-moving standard cameras, the Videojet decoder configuration will be checked for an according rectangular digital zoom setting for that video source. If a preset is found, the according digital zoom settings will be restored.

For Bosch/IQSIGHT panoramic cameras, the Videojet decoder configuration will be checked for an according dewarping digital zoom setting for that video source. If a preset is found, the according digital zoom settings will be restored.

The decoder supports only single digit preset numbers [0..9] for digital zoom.

KBD_STORE_PRESET

	Tag code	NumDes	Message	SNMP Support
	0xe245	yes	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	store preset	

Payload Structure

Monitor Key 2 Bytes	**Preset** 2 Bytes

Monitor Key

NumDes = 0: Monitor table key [0-4095]

NumDes >0: Line = NumDes, coder = monitor key

Preset

Number of preset to be overwritten with the current positioning

For PTZ domes, the preset number is sent to the device with the request to store the current PTZ settings under this number. If presets are not supported by the device or store preset ist not authorized for the camera user, nothing will happen, no error code will be returned.

For all non-moving standard cameras, the current rectangular digital zoom setting will be stored in the Videojet decoder configuration for that video source.

For Bosch/IQSIGHT panoramic cameras, the current dewarping digital zoom setting will be stored in the Videojet decoder configuration for that video source.

The decoder supports only single digit preset numbers [0..9] for digital zoom.

The stored digital zoom settings will all be removed when the digital zoom persistence mode is changed (see DIGITAL_ZOOM_PERSISTENCE_MODE).

KBD_LAYOUT_STEP

	Tag code	NumDes	Message	SNMP Support
	0xe213	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	change the number of video panels	

Payload Structure

Line 2 Bytes	**Layout Step Code** 2 Bytes

Line
Video output line number

Layout Step Code
1: Decrease number of video panels

2: Increase number of video panels

PTZIF

	Tag code	NumDes	Message	SNMP Support
	0xe215	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	manipulate pan, tilt, zoom, iris and focus settings of a video panel	

Payload Structure

Line 1 Byte	Coder 1 Byte	Preset 2 Bytes
Pan Speed 4 Bytes ([-1..1] * 32768 as int32)		
Tilt Speed 4 Bytes ([-1..1] * 32768 as int32)		
Zoom Speed 4 Bytes ([-1..1] * 32768 as int32)		
Iris Speed 4 Bytes ([-1..1] * 32768 as int32)		
Focus Speed 4 Bytes ([-1..1] * 32768 as int32)		

Preset
A preset value of 0 indicates that the speed values shall be applied. A preset value > 0 indicates that the speed values shall be ignored (and may be omitted in the payload), and that the selected PTZ preset shall be activated by the camera.

Speed Values
Speed values between -1.0 and 1.0 are supported. A speed value of 0.12 needs to be set as upscaled int32 speed value (int32)(0.12 * 32768) = (int32)3932,16 = 3932

Multiple speed values can be set in parallel, unused speed values need to be set to zero.

Depending on the displayed video stream and the video source's PTZIF capabilities, either corresponding PTZIF speeds are send to the camera, or mapped to a digital zoom feature (PTZ only), or used for dewarping of panoramic video streams (PTZ only, Bosch/IQSIGHT cameras only).

Digital zoom and dewarping settings will get lost whenever the video source is changed on the respective video panel by default, see DIGITAL_ZOOM_PERSISTENCE_MODE for further options.

Current digital zoom and dewarping states can be retrieved with CONNECT_URL read queries.

FULLSCREEN_TILE

	Tag code	NumDes	Message	SNMP Support
	0xe21b	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	set fullscreen tile and mode	

Payload Structure

Line	Coder	Reserved	Fullscreen Mode
1 Byte	1 Byte	1 Byte	1 Byte

Fullscreen Mode

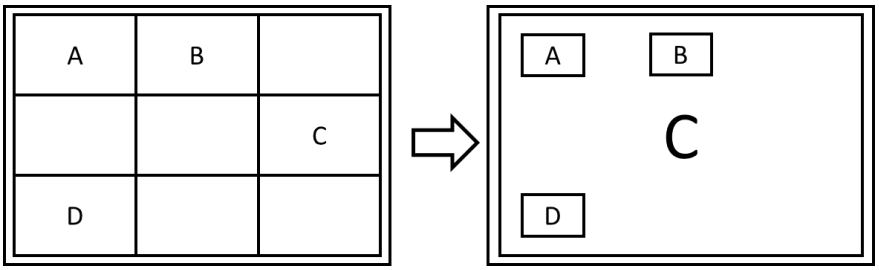
0: Off (coder is ignored)

1: Single fullscreen mode

2: Picture in picture mode

In single fullscreen mode, the selected video panel is maximized and placed in front of the other video panels. All streaming connections are kept open. This is different to 1:1 layout where the video connections are closed for all coders > 1.

In picture in picture mode, the selected video panel is maximized and placed behind all other connected video panels. Not connected video panels will be hidden. All those connected video panels in front of the maximized video panel are scaled down to achieve the typical picture in picture layout. Please note that multiple picture in picture video panels are supported in parallel, and that multiple picture in picture layouts are supported. The actual layout and the position of the connected videos define the picture in picture video panel sizes and positions. One example where video panel C is put into picture in picture mode:



Any video panel that is displaying a video stream can be put into one of the fullscreen modes in any of the supported layouts with any number of other connected video panels. Salvos are paused in fullscreen mode. Fullscreen mode is automatically deactivated on layout changes and new video connections. Active fullscreen modes are persisted in the configuration and applied at startup.

FREEZE_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe216	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	p_octet	user	freeze/unfreeze video panel(s)	

Payload Structure

Line	Coder	Reserved	Freeze Flags
1 Byte	1 Byte	1 Byte	1 Byte

Freeze Flags

Video and metadata overlays can be frozen independently. Set/clear the first bit (0x1) to freeze/unfreeze video, and set/clear the second bit (0x2) to freeze/unfreeze metadata overlays. Both bits can be used in parallel. Salvos will be paused in freeze mode (salvo pause timeout won't apply in this case).

Use line and coder to select a specific video panel, or set line and coder to zero to freeze/unfreeze all video panels.

The freeze state of a video panel will be cleared when a new video is connected.

SALVO_PAUSE_STATE

Tag code	NumDes	Message	SNMP Support
0xe222	no	no	no

	Datatype	Access Level	Description
Read			not supported
Write	t_octet	user	0: continue all active salvos, 1: pause all active salvos. Salvos will stay paused until either the continue command is received or the salvo pause timeout has elapsed (see SALVO_PAUSE_TIMEOUT)

ACTIVE_TILE

Tag code	NumDes	Message	SNMP Support
0xe217	no	no	no

	Datatype	Access Level	Description
Read			not supported
Write	p_octet	user	Select highlighted video panel

Payload Structure Request

Line 1 Byte	**Coder** 1 Byte	**Reserved** 1 Byte	**Highlighting Index** 1 Byte
Reserved 4 Bytes			

Payload Structure Response

Notification 4 Bytes
Reserved 4 Bytes

Highlighting Index

A highlighted video panel is displayed with a different border color. There are 4 different highlighting colors supported, selectable with the highlighting index 1-4. Alarm highlighting (red blinking video panel border) is rendered on top. The same video panel may be highlighted with different highlighting indexes (colors) in parallel, but only the latest one is rendered, former highlightings are (temporarily) hidden.

The following combinations are supported:

Line = 0, coder = 0, highlighting index = 0: Reset highlighting for all video panels

Line = 0, coder = 0, highlighting index 1-4: Reset highlighting for all video panels for specific highlighting index

Line > 0, coder > 0, highlighting index 1-4: Select highlighting color for specific video panel

Notification

The notification in the response is zero whenever the current video panel layout contains the selected coder (video panel). If the selected video panel is not included in the current layout, then the number of video panels in the current layout is returned.

SAVE_FAVORITE

	Tag code	NumDes	Message	SNMP Support
	0xe225	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	t_int32	user	favorite index [0..9]	

Stores the current layout, active connections and active salvos as favorite.

Favorites can be deleted by sending the favorite index with bit 0x10000 set.

Persistence of digital zoom setting is controlled by DIGITAL_ZOOM_PERSISTENCE_MODE.

LOAD_FAVORITE

	Tag code	NumDes	Message	SNMP Support
	0xe21c	no	no	no
	Datatype	**Access Level**	**Description**	
Read			not supported	
Write	t_int32	user	favorite index [0..9]	

Restores a previously saved favorite, i.e. layout, connections and salvos.

Persistence of digital zoom setting is controlled by DIGITAL_ZOOM_PERSISTENCE_MODE.

KBD_LICENSED_CHANNELS

	Tag code	NumDes	Message	SNMP Support
	0xe21f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int32	noprot	reads the number of licensed channels of this decoder for IP matrix keyboard operations	
Write			not supported	

KBD_LICENSED_CHANNELS_GROUP

	Tag code	NumDes	Message	SNMP Support
	0xe223	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int32	noprot (VJD only)	reads the number of licensed channels across the whole decoder group for IP matrix keyboard operations	
Write			not supported	

SERIAL_PORT_APP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x01f1	yes	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user (VJD only)	get current serial port mode	
Write	t_octet	service (VJD only)	set serial port mode	

Numerical descriptor: Serial port number

Serial port modes:

0: Standard mode for Intui keyboards (default)

255: Transparent data mode (see TRANSFER_TRANSPARENT_DATA)

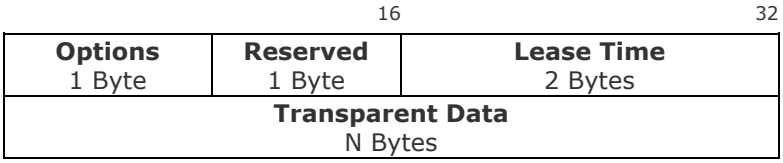
TRANSFER_TRANSPARENT_DATA

	Tag code	NumDes	Message	SNMP Support
	0xffdd	yes	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user (VJD only)	see description	
Write	p_octet	user (VJD only)	see description	

The transparent data from and to the serial interfaces is handled by RCP to achieve reliable transfer of information. To gather control over the remote serial interface a successful registration is necessary. Once the Videojet has received a TRANSFER_TRANSPARENT_DATA command, it checks whether an Intui keyboard is attached or not. If yes, the return code will present an OK (1), else a FAIL (0).
Note: This command is NOT readable in a sense to obtain data from a serial port. In can only be read in order to check if serial port access is currently granted. See further details below. The reply to the read command will be the same as the reply to the write command. The returned code will present the availability of an Intui keyboard.
The data coming from the serial input is delivered using an RCP message. All RCP clients which want to receive this data must be registered for the message 0xffdd.
Note: The received messages will carry NO header fields like the 4 bytes containing options, reserved and lease time.

Numerical descriptor: Serial port number

Payload Structure Write



Options: Ignored
Lease Time: Ignored
Transparent Data: Data to be sent to serial port given by numeric descriptor. Allowed values are 1-4, default is 1.

Payload Structure Response (Read & Write)

32

Code	Reserved
1 Byte	3 Byte

Codes:

0: Either no Intui keyboard available or serial port mode is 0 (see SERIAL_PORT_APP_VAL)

1: Intui keyboard is attached and serial port mode is 255 (see SERIAL_PORT_APP_VAL)

Payload Structure Message

Transparent Data
N Bytes

Transparent Data: Data received at serial port given by numeric descriptor, default is port 1.

Configuration Sealing

CONFIG_SEALING_ENABLED

	Tag code	NumDes	Message	SNMP Support
	0x0c8a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	user	get enabled state of config sealing 0=sealing disabled; 1=sealing enabled; 2=seal broken	
Write	t_octet	service	enable config sealing 0=sealing disabled; 1=sealing enabled	

CONFIG_SEALING_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0c8c	no	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	user	return status of config seal	
Write			read only	

Payload Structure

Status 1 Byte	**Reserved** 3 Bytes
SealSetTimestamp 4 Bytes	
SealRandom 4 Bytes	
SystemTimestamp 4 Bytes	

Status

Values:

Sealing off	0
Sealing on and seal valid	1
Sealing on and seal broken	2

Reserved

Reserved, should be ignored

SealSetTimestamp

Timestamp when seal was activated in seconds since year 2000

SealRandom

Random number generated once when seal was activated

SystemTimestamp

Current system time in seconds since year 2000

Sealing concept

The system can be set up in a way that unexpected configuration changes on the device cause an alert message, even when the user uses a valid login and password for this action. To achieve this: After the whole system configuration is completed, CONFIG_SEALING_ENABLED needs to be set to 1 (= sealing enabled). In this state each critical configuration change causes the device to send a CONFIG_SEALING_STATUS message. In addition, a broken seal will also cause a CONFIG_SEALING_STATUS message at each monitor wall application start.

If a client wants to actively verify if a seal is valid, then it should read CONFIG_SEALING_STATUS and check its content for:

1. Status = 1 (sealing enabled and seal valid)
2. SealSetTimestamp, if the time contained here matches the time of the last known configuration change
3. (Optional) SealRandom, if this number matches the random number of CONFIG_SEALING_STATUS response after enabling the seal
4. SystemTimestamp, if it is about the actual local time (to protect against time manipulation