

# Bosch Monitor Wall RCP+ Reference

---

Version 10.60

## Contents

General Notes .....	6
Glossary.....	7
Transport Protocol .....	8
Remote Control ProtocolPlus Protocol Header Layout.....	8
Identification .....	10
HARDWARE_VERSION .....	10
SOFTWARE_VERSION .....	10
DEVICE_TYPE_IDS.....	11
SERIAL_NUMBER.....	11
UNIT_NAME .....	11
UNIT_ID .....	12
MONITOR_NAME .....	12
SUPPORTED_OPCODES .....	12
CAPABILITY_LIST.....	13
PRODUCT_NAME.....	14
COMMERCIAL_TYPE_NUMBER .....	14
System .....	15
APP_OPTION_UNIT_ID.....	15
APP_OPTION_SET.....	15
PERMISSIONS .....	15
SSD_CAPABILITIES .....	16
SSD_WRITE_PROTECTION_MODE.....	16
SSD_ENCRYPTION_MODE .....	17
USB_MODE.....	17
PTZIF_SENSITIVITY.....	18
OVERLOAD_STATUS .....	19
Network.....	20

MAC\_ADDRESS.....20

IP\_STR.....20

GATEWAY\_IP\_STR .....20

SUBNET\_STR.....20

DEVICE\_IP\_ADDRESS.....21

DHCP\_VAL .....21

DNS\_SERVER\_IP\_STRING .....22

HOST\_NAME.....22

Time.....23

DATE\_WDAY.....23

DATE\_DAY .....23

DATE\_MONTH.....23

DATE\_YEAR.....23

TIME\_HRS.....24

TIME\_MIN .....24

TIME\_SEC.....24

TIMEZONE .....24

TIME\_ZONE\_STRING .....25

NTP\_SERVER\_IP\_STR.....25

FORCE\_TIME\_SET .....25

SYSTEM\_DATETIME\_V2.....25

Connection .....26

PASSWORD\_SETTINGS .....26

ALARM\_CONNECT\_TO\_IP\_STR .....26

REMOTE\_PASSWORD.....26

NBR\_OF\_ALTERNATIVE\_ALARM\_IPS.....27

ALARM\_CONNECT\_TO\_IP .....27

CONNECT\_TO .....27

DISCONNECT\_PRIMITIVE.....29

ACTIVE\_CONNECTION\_LIST.....30

CONNECT\_URL .....31

VIDEO\_OUTPUT\_STATUS .....33

DISCOVERY\_PORT.....33

- RECONNECT.....34
- Monitor .....35
  - BOARD\_RESET .....35
  - DECODER\_LAYOUT\_LIST .....35
  - DECODER\_LAYOUT .....37
  - CAMEO\_DISTANCE .....37
  - CAMEO\_ALARM\_MODE .....38
  - SWAP\_MONITOR.....38
  - SHOW\_METADATA.....39
  - OVERLAY\_MODE .....41
  - AUTO\_RESOLUTION\_CHANGE .....41
  - TARGET\_DISPLAY\_ORIENTATION .....41
  - SMOOTHNESS.....42
  - RENDERING\_MODE .....42
  - PREFERRED\_ASPECT\_RATIO.....44
  - NUMBER\_OF\_DECODERS .....45
  - ZOOM\_MODE.....45
  - DIGITAL\_ZOOM\_PERSISTENCE\_MODE .....45
  - MONITOR\_INFO .....46
  - MONITOR\_INFO2 .....47
  - MONITOR\_POWER\_MODE.....48
  - VOCS\_CONFIG .....48
  - Snapshot download.....48
  - Background logo up- and download .....50
  - LOGO\_LIST.....50
- Banners .....51
  - Banner up- and download.....51
  - BANNER\_LIST.....51
  - CONFIG\_BANNER\_SALVO.....51
  - CONFIG\_BANNER\_SALVOS.....52
  - CONNECT\_BANNER\_SALVO .....53
- Stamping .....55
  - NAME\_STAMP\_VAL.....55

TIME\_STAMP\_VAL.....55

ALARM\_DISP\_VAL .....55

STAMP\_ATTR\_NAME.....56

STAMP\_ATTR\_TIME.....57

STAMP\_ATTR\_ALARM.....57

ALARM\_STRING.....57

Keyboard and IP Matrix Configuration.....58

  KBD\_CONFIG\_CAMERA\_STR .....58

  KBD\_CONFIG\_MONITOR\_STR.....59

  KBD\_CONNECT\_PARAMS.....59

  KBD\_PASSWORD\_CAMERA.....60

  KBD\_PASSWORD.....60

  KBD\_TIMEOUT.....60

  KBD\_CONFIG\_SALVO.....60

  KBD\_CONFIG\_SALVO2.....61

  KBD\_CONFIG.....62

  DECODER\_GROUP.....63

  FREEZE\_CHECK\_INTERVAL.....64

  SALVO\_PAUSE\_TIMEOUT.....64

Keyboard and IP Matrix Operations.....65

  INFO\_OVERLAY\_MODE.....65

  KBD\_CONNECT\_CAMERA.....65

  KBD\_CONNECT\_SALVO.....65

  KBD\_GOTO\_PRESET.....66

  KBD\_STORE\_PRESET.....67

  KBD\_LAYOUT\_STEP.....67

  PTZIF.....68

  FULLSCREEN\_TILE.....69

  FREEZE\_MODE.....70

  SALVO\_PAUSE\_STATE.....71

  ACTIVE\_TILE.....71

  SAVE\_FAVORITE.....72

  LOAD\_FAVORITE.....72

KBD\_LICENSED\_CHANNELS .....72

KBD\_LICENSED\_CHANNELS\_GROUP .....72

SERIAL\_PORT\_APP\_VAL .....73

TRANSFER\_TRANSPARENT\_DATA .....73

Configuration Sealing .....75

CONFIG\_SEALING\_ENABLED .....75

CONFIG\_SEALING\_STATUS.....75

## General Notes

### Copyright

This manual is the intellectual property of Bosch Security Systems 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 Bosch Security Systems.  
© Copyright 2001-2023 Bosch Security Systems

### 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. Bosch Security Systems will not be liable for damage which is directly or indirectly due to errors, incompleteness, or discrepancies between the manual and the product described.

### Trade marks

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

## Glossary

Hardware Decoder	Physical device that hosts one instance of the Monitor Wall application.
Monitor Wall	The name of the main application designed for showing streams (video, audio and metadata) from different video sources.
Line	One-based index for physical video out connectors of a hardware decoder. Each line corresponds thus to one connected physical screen.
Coder	One-based index for video panels on each line. Each video panel can display one video stream.
Layout	Arrangement of video panels (coders) on a line. Each line can have its own layout.
IP Matrix	Cluster of 1 to 4 hardware decoders that control up to 16 physical screens.
VOCS	Visual output capture service

# Transport Protocol

The transport protocol for this remote control follows the general specifications for Bosch Video over IP cameras. For further details, please refer to the corresponding Rcp+ specification document.

## Remote Control ProtocolPlus Protocol Header Layout

Version 3 (VIP, VideoJet and VipX Series and VIP110Version 6.0 and higher). The RCP Plus protocol header consists of 16 Bytes. The begin of the payload section is now on DWORD boundary.

<b>Tag</b> 2 Bytes				<b>Data Type</b> 1 Byte	<b>Version</b> 4 Bits	<b>R/W</b> 4 Bits
<b>C</b> 1	<b>T</b> 1	<b>Action</b> 6 Bits	<b>Reserved</b> 1 Byte	<b>Client ID</b> 2 Bytes		
<b>Session ID</b> 4 Bytes						
<b>Numeric Descriptor</b> 2 Bytes				<b>Payload Length</b> 2 Bytes		

### 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

StringTable 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 .

## Identification

### HARDWARE\_VERSION

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002e	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	noprot	read the hardware version	
<b>Write</b>			not supported	

Gets the system hardware version.

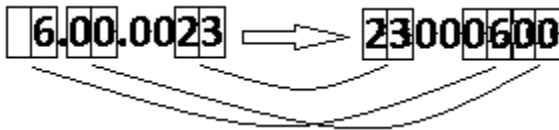
- 0xf1004940 VIP XD HD (discontinued)
- 0xf1005440 VJD 7000 (discontinued)
- 0xf1009040 Videojet decoder family (VJD 8000, VJD 7513, VJD 7523), see DEVICE\_TYPE\_IDS
- 0xf100fe40 Monitor Wall stand-alone software

### SOFTWARE\_VERSION

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002f	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	Noprot	read the software version	
<b>Write</b>			not supported	

Gets the system software version.

For example: For build 6.00.0023 result is 23000600:



**DEVICE\_TYPE\_IDS**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0b07	No	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	Noprot	read the device type ids	
<b>Write</b>			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, VJD 7513, VJD 7523)
- 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

3rd four bytes: reserved

**SERIAL\_NUMBER**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0ae7	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	Noprot	returns the serial number of the device	
<b>Write</b>			not supported	

Gets the system serial number.

If it is not defined the result is "000000000000000000". Max length is 127characters.

**UNIT\_NAME**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0024	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_unicode	Noprot	read the unit name	
<b>Write</b>	p_unicode	Service	set unit name	

Specifies the unit name(max 32 unicode characters).

### UNIT\_ID

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0025	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_unicode	Noprot	read the unit ID	
<b>Write</b>	p_unicode	Service	set unit ID	

Specifies the unit name(max 32 unicode characters).

### MONITOR\_NAME

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x028a	Line	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_unicode	noprot	read the video monitor name	
<b>Write</b>	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

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

Gets the list of supported opcodes for this session.

### Reply payload Structure

<b>OpCode 1</b> 4 Byte
... 4 Bytes
<b>OpCode N</b> 4 Byte

## CAPABILITY\_LIST

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

Gets the list of system capabilities.

### Reply payload Structure

<b>0xBABA</b> 2 Bytes	<b>Version</b> 2 Bytes	<b>NbSection</b> 2 Bytes	<b>Section 1</b>	...	<b>Section N</b>
--------------------------	---------------------------	-----------------------------	------------------	-----	------------------

#### Version

current version of the capabilities (0x0001)

#### NbSection

Number of following sections. Hardcoded (0x0019)

### Section Structure

<b>Type</b> 2 Bytes	<b>Size</b> 2 Bytes	<b>NbElement</b> 2 Bytes	<b>Element 1</b>	...	<b>Element N</b>
------------------------	------------------------	-----------------------------	------------------	-----	------------------

#### 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 using the size.

#### NbElement

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

### Element Structure

<b>Type</b> 2 Bytes	<b>Identifier</b> 2 Bytes	<b>Compr</b> 2 Bytes	<b>InputNo</b> 2 Bytes	<b>Resolution</b> 2 Bytes
------------------------	------------------------------	-------------------------	---------------------------	------------------------------

#### Type

is one of the following

Values:

VIDEO\_DECODER 0x0002

#### Identifier

One-based coder number.

#### Compression

is one or multiple of the following

Values:

VIDEO_COMP_MPEG2	0x0001
VIDEO_COMP_MPEG4	0x0002
VIDEO_COMP_H264	0x0004
VIDEO_COMP_JPEG	0x0008
VIDEO_COMP_H265	0x0010

**OutputNo**

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	
<b>Read</b> p_string	noprot	Read the commercial type number (CTN) of the device (SKU number).	
<b>Write</b>		not supported	

**COMMERCIAL\_TYPE\_NUMBER**

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

## System

### APP\_OPTION\_UNIT\_ID

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x09e1	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	User	read unique unit id (installation code)	
<b>Write</b>			not supported	

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

### APP\_OPTION\_SET

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x09e2	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	User	read installed application options	
<b>Write</b>	p_string	Service		

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 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 hardware decoders may include pre-installed application options that cannot be deleted, identifiable by authorization numbers like "BoschDevice".

### PERMISSIONS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe22a	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	User	read permissions	
<b>Write</b>	t_int	Service	write permissions	

Gets and sets the permissions as bit-field (only the lower 16 bit are used):

Bit 0 (0x1): 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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe226	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	User	read the SSD capabilities	
<b>Write</b>			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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe227	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	User	read the SSD write protection mode	
<b>Write</b>	t_word	service (VJD only)	set the SSD write protection mode	

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 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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe228	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	User	read the SSD encryption mode	
<b>Write</b>	t_word	service (VJD only)	set the SSD encryption mode	

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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe229	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	User	read the USB lockdown mode	
<b>Write</b>	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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe247	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	User	read PTZIF sensitivity	
<b>Write</b>	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. jetplane 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 begin of 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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe248	no	yes	yes
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_dword	user	read or get notification on overload status	
<b>Write</b>		service	not supported	

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

## Network

### MAC\_ADDRESS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x00bc	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	Noprot	read out the systems MAC address	
<b>Write</b>			not supported	

Gets the system MAC address.

### IP\_STR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x007c	no	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	noprot	read the unit's IP address using string notation (xxx.xxx.xxx.xxx)	
<b>Write</b>	p_string	service (VJD only)	set unit's IP address using string notation (xxx.xxx.xxx.xxx)	

Specifies the system IP address

Note: Monitor Wall should be started with administrative rights.

### GATEWAY\_IP\_STR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x007f	no	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	noprot	read the gateway IP using string notation (xxx.xxx.xxx.xxx)	
<b>Write</b>	p_string	service (VJD only)	set gateway IP using string notation (xxx.xxx.xxx.xxx)	

Specifies the system gateway IP address.

Note: Monitor Wall should be started with administrative rights.

### SUBNET\_STR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x007d	no	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	noprot	get current subnet mask string notation (xxx.xxx.xxx.xxx)	
<b>Write</b>	p_string	service (VJD only)	set subnetmask string notation (xxx.xxx.xxx.xxx)	

Specifies current system subnet mask.

Note: Monitor Wall should be started with administrator rights

**DEVICE\_IP\_ADDRESS**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xd052	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	Noprot	get network configuration	
<b>Write</b>	p_octet	service (VJD only)	set current network configuration	

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:

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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x00af	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	noprot	read the DHCP state (0=OFF, 1=ON)	
<b>Write</b>	t_octet	service (VJD only)	set DHCP ip configuration (0=OFF, 1=ON)	

**DNS\_SERVER\_IP\_STRING**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0b49	Yes	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	Noprot	get DNS server IPv4	
<b>Write</b>	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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0cda	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_unicode	Noprot	read the hostname	
<b>Write</b>	p_unicode	service (VJD only)	set host name (max 128 unicode character)	

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

# Time

## DATE\_WDAY

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

Gets the current system weekday.

For example if date is 01/17/2013, result will be "Thursday".

## DATE\_DAY

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0028	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	Noprot	read the day of month	
<b>Write</b>			not supported	

Gets the current system day.

Example: Date is 1/17/2013. Result is 17

## DATE\_MONTH

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0029	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	Noprot	read the month	
<b>Write</b>			not supported	

Gets the current system month.

Example: Date is 1/17/2013. Result is 1

## DATE\_YEAR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002a	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	Noprot	read the year	
<b>Write</b>			not supported	

Gets the current system year;

Example: Date is 1/17/2013. Result is 2013

### TIME\_HRS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002d	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	Noprot	read the hours	
<b>Write</b>			not supported	

Gets the current system count of hours;

Example: Time is 14:35:48. Result is 14

### TIME\_MIN

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002c	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	Noprot	read the minutes	
<b>Write</b>			not supported	

Gets the current system count of minutes.

Example: Time is 14:35:48. Result is 35

### TIME\_SEC

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x002b		no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	Noprot	read the seconds	
<b>Write</b>			not supported	

Gets the current system count of seconds.

Example: Time is 14:35:48. Result is 48

### TIMEZONE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x024e	no	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	noprot	the timezone in which the unit has to operate (UTC +- nbr of seconds +- nbr of seconds DLS)	
<b>Write</b>			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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xd027	no	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	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.	
<b>Write</b>	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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x024f	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	Noprot	Get the NTP server URL	
<b>Write</b>	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 be part of the URL (always 123). If provided, the NTP server URL will be automatically clipped to remove the port ":123".

**FORCE\_TIME\_SET**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0a0f	No	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	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)	
<b>Write</b>	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)	

**SYSTEM\_DATETIME\_V2**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0ba8	No	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	noprot	Gets the UTC system date and time: yyyy-mm-ddThh:mm:ss;; Timezone can only be retrieved via TIME_ZONE_STRING	
<b>Write</b>	p_string	service (VJD only)	Writes the UTC system date and time: yyyy-mm-ddThh:mm:ss;; Timezone can only be changed via TIME_ZONE_STRING	

## Connection

### PASSWORD\_SETTINGS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x028b	password level	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	service, user	get the unit password (scrambled). num parameter sets the password levels; 1=user, 2=service	
<b>Write</b>	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
- service

### ALARM\_CONNECT\_TO\_IP\_STR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0081	Password table index	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	service, user	read IP using string notation (xxx.xxx.xxx.xxx)	
<b>Write</b>	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x010c	Password table index	No	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	void	service, user	password	
<b>Write</b>	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 cameras, Bosch encoders and Bosch 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

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

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

### ALARM\_CONNECT\_TO\_IP

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0041	destination IP number	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_dword	service, user	specifies the IP address .	
<b>Write</b>	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 cameras, Bosch encoders and Bosch 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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xffcc	No	yes	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>			not supported	
<b>Write</b>	p_octet	service, user	see detailed description	

Connects media streams from a device. For password-protected Bosch cameras, Bosch encoders and Bosch 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

<b>Destination IP Address</b>		
4 Bytes		
<b>Reserved</b> 1 Byte	<b>Reserved</b> 1 Byte	<b>Flags</b> 2 Bytes
<b>Local Coder</b> 1 Byte	<b>Local Line</b> 1 Byte	<b>Put Channels</b> 2 Bytes
<b>Remote Coder</b> 1 Byte	<b>Remote Line</b> 1 Byte	<b>Get Channels</b> 2 Bytes

### Destination IP Address

The camera’s or encoder’s IP address.

**Flags**Values:

Bit0	Reserved
Bit1	Request video mode MPEG-4
Bit2	Request video mode MPEG-2
Bit3	Force the use of TCP as transportation protocol
Bit4	Connect to the HDD to receive recorded stream
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.
Bit1	Audio
Bit2	Data. Ignored in this command.
Bit3-Bit15	Reserved

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

**Optional Appendix**

<b>Destination Port</b> 2 Bytes	<b>Reserved</b> 2 Bytes
<b>Reserved</b> 4 Bytes	

**Destination Port**

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

**Reply / Read / 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**

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

Disconnecting video streams from the application

**Reply Payload Structure**

<b>Status</b> 1 Byte	<b>Cause</b> 1 Byte	<b>Reserved</b> 2 Bytes
<b>Remote Host IP</b> 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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xffc1	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	see detailed description	
<b>Write</b>			not supported	

Gets list of all active connections.

**Payload Structure**

Sequence of:

<b>Destination IP Address</b> 4 Bytes		
<b>Local Coder</b> 1 Byte	<b>Local Line</b> 1 Byte	<b>Flags</b> 2 Bytes
<b>Session ID</b> 4 Bytes		
<b>Remote Coder</b> 1 Byte	<b>Remote Line</b> 1 Byte	<b>Destination Port</b> 2 Bytes
<b>TX Channels</b> 4 Bytes		
<b>RX Channels</b> 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:

Bit0	Connection is MPEG2 VES
Bit2	Connection is MPEG4
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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20c	Read request only	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	user	URL with optional parameters	
<b>Write</b>	p_string	service, user	URL with optional parameters	

Connect a video panel to an url. For a list of supported urls please check the VideoSDK documentation. 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): screen in upper 8 bit, tile in lower 8 bit

String format (separator: space character):

<url> [<option1>=<value1>] ... [<optionN=<valueN>]

## Optional parameters:

screen=<1..>	Monitor Wall line (default: 1)
tile=<1..>	Monitor Wall coder (default: 1)
line=<1..>	Camera line number (default = 1). Not supported for some urls.
coder=<1..>	Camera encoder number (default = 1). Not supported for some urls.
protocol=<tcp udp>	Streaming protocol (default see VideoSDK documentation)
multicast=<true false>	Multicast flag (default: true for rcpp urls, else false)
type=<string>	VideoSDK proxy type, e.g. VIP (for Bosch cameras), RTSP, ONVIF, etc.
audio=<uint>	Audio flags (0x1: connect audio stream)
preset=<int>	Dome preset to be applied after successful connection
centerX=<float 0..1> centerY=<float 0..1> scale=<float>	VideoSDK IZoomView parameters
pan=<float> tilt=<float> scale=<float>	VideoSDK IDewarpingView parameters
recording=<uint>	Recording flags (0x1: connect to HDD recording)

## Examples:

rcpp://user:\*@192.168.1.2

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



**VIDEO\_OUTPUT\_STATUS**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20e	No	yes	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	noprot	Not supported, message only	
<b>Write</b>			Not supported	

Signals changes of the video output

**Payload Structure**

<b>Local Coder</b>	<b>Local Line</b>	<b>Status</b>
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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0976	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_word	service, user	reading current value of port	
<b>Write</b>	t_word	service	set port for application discovery via multicast	

Specifies the discovery port.

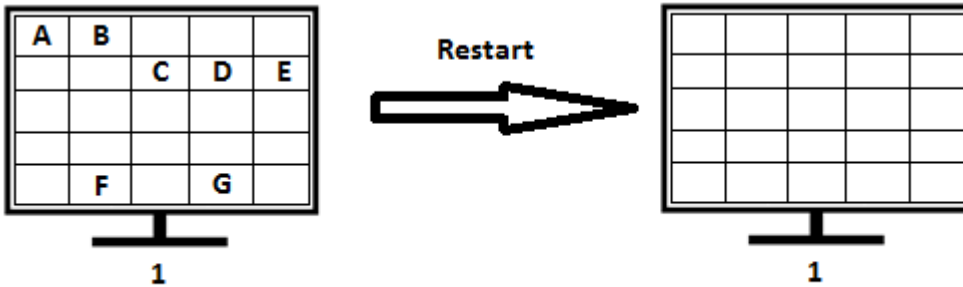
NOTE: Default value is 1800.

**RECONNECT**

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

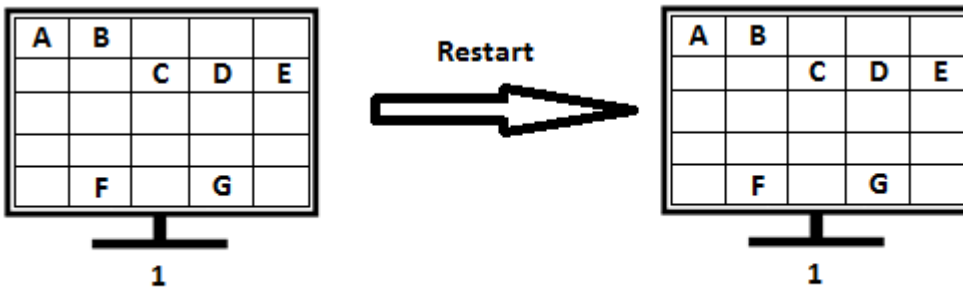
Specifies whether all streams should be reconnected after application restart.

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



A-G – connected video streams

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



# Monitor

## BOARD\_RESET

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0811	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>			not supported	
<b>Write</b>	void	service	Resets the board.	

This command restarts the Monitor Wall application after returning the result.

Write command returns 0x01 value.

## DECODER\_LAYOUT\_LIST

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x09a1	Line	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	list of supported layouts. Each element has 2 bytes	
<b>Write</b>			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 in detail describes the Monitor Wall consisting of an array 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 deviating from the camera 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 four 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.

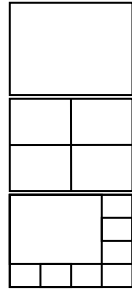
For example for a monitor with optimal resolution 1920 X 1080 the following layouts are supported. Other resolutions may have different values. Any monitor resolutions are supported.

Aspect ratio	Layouts										
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									
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	
<b>Read</b>	p_octet	service, user	first 2bytes are the layout, followed by the coder list in bytes (000101 for first monitor singleview, 000201020304 for quadview with chronological order)	
<b>Write</b>	p_octet	service, user	First 2 bytes are the layout, all further bytes will be ignored	

Gets the current layout for specified line.

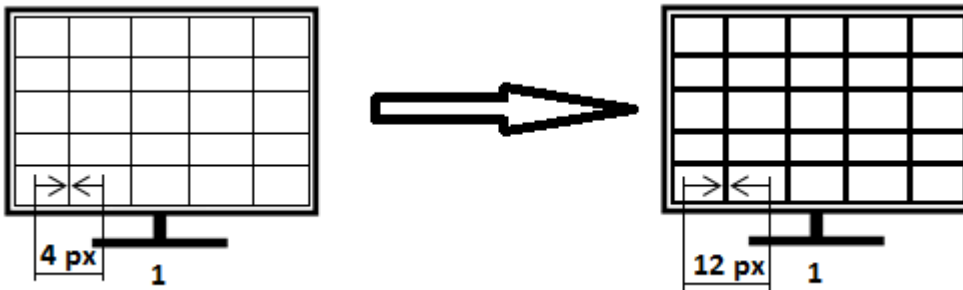
Command uses the same layout codes as DECODER\_LAYOUT\_LIST command

### CAMEO\_DISTANCE

	Tag code	NumDes	Message	SNMP Support
	0xe209	no	no	no
	Datatype	Access Level	Description	
<b>Read</b>	t_int	service, user	get current value	
<b>Write</b>	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 and 6px:



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

### CAMEO\_ALARM\_MODE

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

Specifies the video panel border behavior in case of an alarm event.

- 0: Alarms are suppressed
  - 1: Alarms are displayed as blinking red border around the respective video panel for some seconds
  - 2: Alarms are displayed as static red border around the respective video panel for some seconds
- Default value is 1.

### SWAP\_MONITOR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe207	No	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	get current screen order	
<b>Write</b>	p_octet	Service	see detailed description	

Specifies the mapping between physical video outputs and line numbers. The payload consists of the one-based physical video output number for line 1 (1 byte), followed by the one-based the physical video output number for line 2 (1 byte), and so on, for each supported line (independent of the number of actual active lines with attached monitors). All values in the payload must be unique and between one and the maximum number of supported lines.

### Payload structure

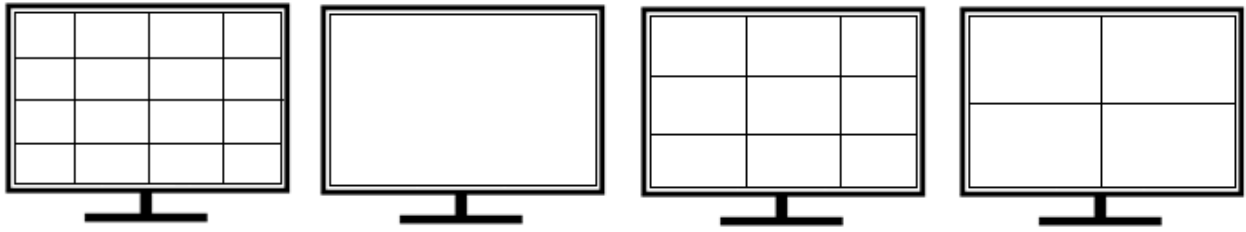
<b>One-based physical video output number for line 1</b> 1 Byte	<b>... for line 2</b> 1 Byte	...
--	---------------------------------	-----

In order to achieve a certain desired order of monitors, the mapping between line number and physical video output 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), send four bytes {0x01, 0x02, 0x03, 0x04} as payload

With that, line 1 maps to physical video output 1, line 2 maps to physical video output 2, and so on.

Then, as **second step**, the line number of each 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).



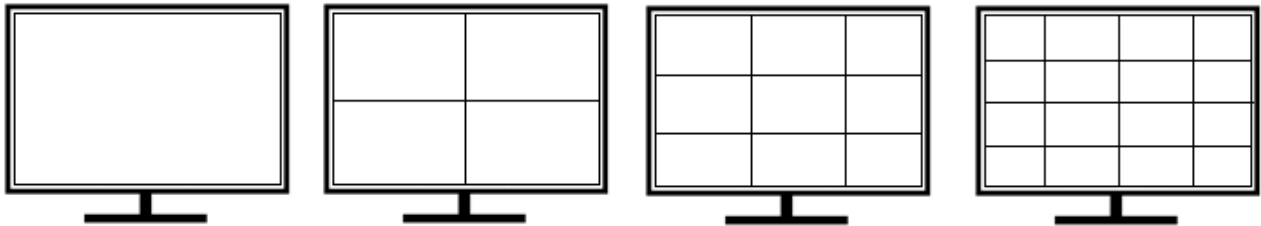
The one-based physical video output numbers, from left to right, for the monitor arrangement in the example above are given by: 4 (Line 4), 1 (Line 1), 3 (Line 3), 2 (Line 2)

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 send in this example as **third step**: {0x04, 0x01, 0x03, 0x02}

The payload matches the identified order of physical video outputs from left to right.

Note: This command restarts Monitor Wall after returning the result of a write command.

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



### SHOW\_METADATA

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe201	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	f_flag	service, user	get current value	
<b>Write</b>	f_flag	service	set rendering of VCA overlay	

Specifies whether movement metadata should be shown.

This command allows the Monitor Wall to show visual overlays if there is any movement on video stream.

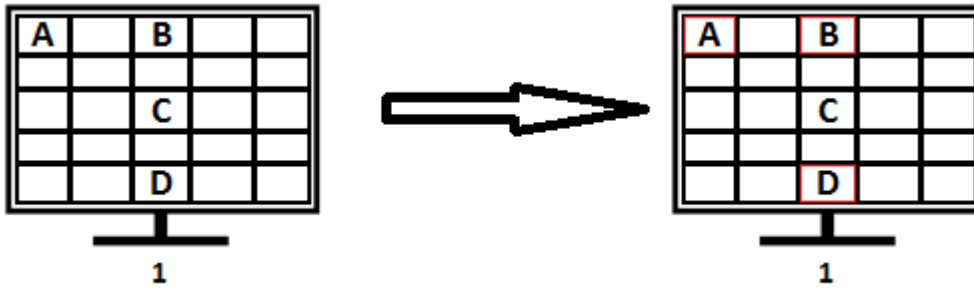
There are two visual effects:

- Blinking red border
- Video content analysis (VCA) overlay

Parameter value: 1 – Enable; otherwise – Disable.

Red border is the default visual effect; VCA overlay depend on VCA meta data (see CONNECT\_TO command).

Example of red borders:



A-D – connected video streams.

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



## OVERLAY\_MODE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe243	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	service, user	get current overlay mode	
<b>Write</b>	t_int	service (VJD only)	set overlay mode	

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 (see chapter **Banners**)
- 3 reserved
- 4 deprecated

## AUTO\_RESOLUTION\_CHANGE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe204	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	f_flag	service, user	get current value	
<b>Write</b>	f_flag	service (VJD only)	set automatic monitor resolution	

NOTE: The datatype is f\_flag, but will be interpreted as t\_octet (both datatypes have the size 1 byte).

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 compatible with 60Hz video (30Hz or 60Hz screen frequency)
- 9: Highest screen resolution compatible with 50Hz video (50Hz or 75Hz screen frequency)

## TARGET\_DISPLAY\_ORIENTATION

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe23d	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	get current value(s)	
<b>Write</b>	p_octet	service (VJD only)	set target display orientation(s)	

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 MonitorWall 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:

<b>Line</b> 1 Byte	<b>Target Display Orientation</b> 1 Byte
-----------------------	---

Read: Payload contains one two-byte-entry per attached display

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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe202	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	service, user	get current value	
<b>Write</b>	t_int	service	set smoothness level for jittery video connections [0..100]	

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 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.

## RENDERING\_MODE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe24d	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_dword	service, user (VJD-7523 only)	get current value	
<b>Write</b>	t_dword	service (VJD-7523 only)	set rendering mode	

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-7523 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-7523 datasheet for more details). For some use case 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-7523 will then operate outside its specification, leading to a lower 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`.

### PREFERRED\_ASPECT\_RATIO

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe203	no	no	No
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	service, user	get current value	
<b>Write</b>	t_octet	service	set current aspect ratio. 0='4:3'; 1='16:9'; 2='9:16'; 3='3:4'	

Specifies the current aspect ratio.

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

The standard values zero to three are defined as enumeration:

0='4:3';

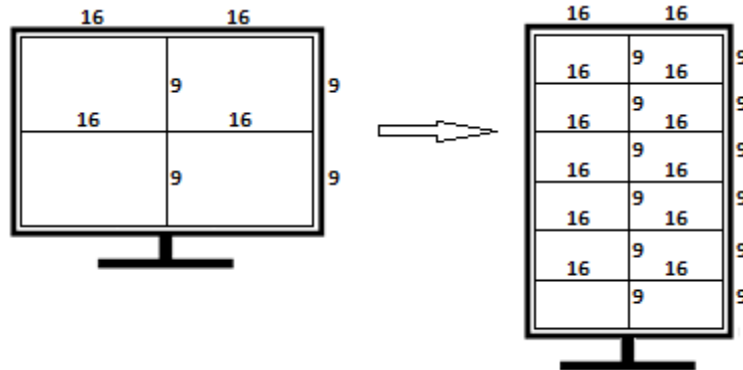
1='16:9';

2='9:16';

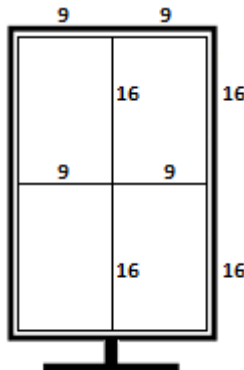
3='3:4'

Any values above are defined as width percentage of the height. So 4:3 can also be defined as 133.

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

## NUMBER\_OF\_DECODERS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20b	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	service, user	get current limit	
<b>Write</b>	t_int	service	set current limit	

Specifies the maximum number of video panels across all lines supported by the device. The limit per line is obtained by dividing the value by the number of lines.

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 device operation a reboot of the device and optionally the client may be required.

## ZOOM\_MODE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20c	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	service, user	get current value	
<b>Write</b>	t_int	service	set mode	

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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe23e	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	service, user	get current value	
<b>Write</b>	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, camera encoder index). 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, camera encoder index) 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).

**MONITOR\_INFO**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20d	Line	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	see detailed description	
<b>Write</b>			not supported	

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

**Info Structure**

<b>Width</b> 4 Bytes	<b>Height</b> 4 Bytes	<b>RefreshRate</b> 4 Bytes
-------------------------	--------------------------	-------------------------------

**Width**

Horizontal number of display pixel

**Height**

Vertical number of display pixel

**RefreshRate**

Display refresh rate in Hz

**MONITOR\_INFO2**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe22d	Line	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	see detailed description	
<b>Write</b>			not supported	

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

**Info Structure**

<b>Left</b>	<b>Top</b>	<b>Width</b>	<b>Height</b>	<b>RefreshRate</b>
4 Bytes	4 Bytes	4 Bytes	4 Bytes	4 Bytes

**Left**

Horizontal position in display pixel

**Top**

vertical position in display pixel

**Width**

Horizontal number of display pixel

**Height**

Vertical number of display pixel

**RefreshRate**

Display refresh rate in Hz

## MONITOR\_POWER\_MODE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe20f	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int	service, user	get current value	
<b>Write</b>	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe233	Either line or 0 for all lines	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	get current config	
<b>Write</b>	p_octet	service (VJD-7513 and newer only)	set config	

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.  $\log_2(4) = 2$ ).

Numerical descriptor: 0

Payload format (6 Bytes per line):

<b>Line</b>	<b>Frame rate divisor (log2)</b>	<b>Size divisor (log2)</b>
2 Bytes	2 Bytes	2 Bytes

Numerical descriptor: Line

Payload format (4 Bytes):

<b>Frame rate divisor (log2)</b>	<b>Size divisor (log2)</b>
2 Bytes	2 Bytes

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

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

## Snapshot download

Screen and tile snapshots 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 during a snapshot request.

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

**JpegScreen=<1..maxScreen>**

**JpegTile=<1..maxTile>**

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>

**JpegQuality=<1..32>**

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

Background logos can be uploaded via http PUT /Logo.jpg and http PUT /NoCamLogo.jpg, download is supported via http GET /Logo.jpg and http GET /NoCamLogo.jpg.

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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe243	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_string	service, user	get current list of active logos	
<b>Write</b>	p_string	service	set list of logos that shall remain active	

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

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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe241	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	get current list of uploaded banners	
<b>Write</b>	p_octet	service (VJD only)	set list of banners that shall remain available	

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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe23f	Banner salvo identifier [1-16]	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	

<b>Read</b>	p_octet	service, user	read configuration of banner salvo
<b>Write</b>	p_octet	service (VJD only)	configure banner salvo

### Payload Structure

<b>Cycle Time</b> 2 Bytes	<b>Reserved</b> 2 Bytes
<b>Banner Identifier 1</b> 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	
<b>Read</b>	p_octet	service, user	read whole banner salvo configuration	
<b>Write</b>	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 filter 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

<b>Length</b> 2 Bytes	<b>Type</b> 2 Bytes	<b>Banner Salvo Identifier</b> 2 Bytes	<b>Reserved</b> 2 Bytes	<b>Banner Salvo Configuration</b> 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-8 bytes with the actual banner salvo configuration.

Banner salvo type 0x0001: See CONFIG\_BANNER\_SALVO for payload definition

## CONNECT\_BANNER\_SALVO

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe240	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	get overview of banner salvo(s)	
<b>Write</b>	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:

<b>Line</b> 2 Bytes	<b>Position</b> 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

<b>Line</b> 2 Bytes	<b>Position</b> 2 Bytes	<b>Banner Salvo Identifier</b> 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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0084	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	service, user	get current value	
<b>Write</b>	t_octet	service (VJD only)	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0085	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	service, user	get current value	
<b>Write</b>	t_octet	service (VJD only)	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x008e	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_octet	service, user	get current value	
<b>Write</b>	t_octet	service (VJD only)	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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0936	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	Set configuration	
<b>Write</b>	p_octet	service (VJD only)	Get configuration	

<b>x</b> 1 Byte	<b>y</b> 1 Byte	<b>reserved</b> 2 Bytes	
<b>attributes</b> 4 Byte			
<b>reserved</b> 2 Bytes		<b>flags</b> 2 Bytes	
<b>Color</b> 4 Byte			
<b>BackgroundColor</b> 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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0937	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	Set configuration	
<b>Write</b>	p_octet	service (VJD only)	Get configuration	

For the payload definition refer to STAMP\_ATTR\_NAME.

### STAMP\_ATTR\_ALARM

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0938	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	Set configuration	
<b>Write</b>	p_octet	service (VJD only)	Get configuration	

For the payload definition refer to STAMP\_ATTR\_NAME.

### ALARM\_STRING

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0090	yes	no	yes
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_unicode	service, user	read the alarm string of a cameo	
<b>Write</b>	p_unicode	service (VJD only)	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:

<b>Line</b>	<b>Coder</b>
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	
<b>Read</b>	p_octet	service, user	read entry of camera table	
<b>Write</b>	p_octet	service (VJD only)	configure camera table entry	

### Payload Structure

Line	Coder	Preset	URL Length
1 Byte	1 Byte	1 Byte	1 Byte
<b>URL and Options</b> 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)

line=<1..>	Camera line number (default = 1). Not supported for some urls.
coder=<1..>	Camera encoder number (default = 1). Not supported for some urls.
type=<string>	VideoSDK proxy type, e.g. VIP (for Bosch cameras), RTSP, ONVIF, etc.
audio=<uint>	Audio flags (0x1: connect audio stream)
preset=<int>	Dome preset to be applied after successful connection
centerX=<float 0..1> centerY=<float 0..1> scale=<float>	VideoSDK IZoomView parameters
pan=<float> tilt=<float> scale=<float>	VideoSDK IDewarpingView parameters

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

### KBD\_CONFIG\_MONITOR\_STR

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0ba4	Monitor table key [0-4095]	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	read monitor table entry	
<b>Write</b>	p_octet	service (VJD only)	configure monitor table entry	

### Payload Structure

<b>Line</b> 1 Byte	<b>Coder</b> 1 Byte	<b>URL Length</b> 1 Byte	<b>Reserved</b> 1 Byte
<b>URL</b> 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: One byte with index of the decoder in the decoder group [0-3], see DECODER\_GROUP

### KBD\_CONNECT\_PARAMS

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0a33	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	read parameters for connections established via camera table	
<b>Write</b>	p_octet	service (VJD only)	set parameters for connections established via camera table	

### Payload Structure

<b>Flags</b> 2 Bytes	<b>Reserved</b> 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

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

### KBD\_PASSWORD

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

### KBD\_TIMEOUT

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0a6a	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int32		read keyboard lock timeout in minutes	
<b>Write</b>	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0x0a3e	Salvo table key [1-32]	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	read configuration of salvo from salvo table	
<b>Write</b>	p_octet	service (VJD only)	configure salvo in salvo table	

### Payload Structure

<b>Duration</b> 2 Bytes	<b>Line</b> 1 Byte	<b>Coder</b> 1 Byte
<b>Camera Table Key 1</b> 1 Byte	N x 1 Byte	<b>Camera Table Key N</b> 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
Read	Datatype	Access Level	Description
	p_octet	service, user	read configuration of salvo from salvo table
Write	p_octet	service (VJD only)	configure salvo in salvo table

**Payload Structure**

<b>Duration</b> 2 Bytes	<b>Line</b> 1 Byte	<b>Coder</b> 1 Byte
<b>Camera Table Key 1</b> 2 Bytes	N x 2 Bytes	<b>Camera Table Key N</b> 2 Bytes

**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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe21a	Table entry type selection	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	read configuration tables	
<b>Write</b>	p_octet	service (VJD only)	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

<b>Length</b> 2 Bytes	<b>Type</b> 2 Bytes	<b>Key</b> 2 Bytes	<b>Reserved</b> 2 Bytes	<b>Entry</b> 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 payload sections of the commands KBD\_CONFIG\_CAMERA\_STR, KBD\_CONFIG\_MONITOR\_STR, and KBD\_CONFIG\_SALVO2 (usage 1).

## DECODER\_GROUP

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe21d	No	no	No
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	p_octet	service, user	read decoder group configuration	
<b>Write</b>	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 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)
- 4 bytes per supported line (VJD-7513: 2 supported lines; VJD-7523: 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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe220	no	no	No
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int32	noprot	read freeze check interval in seconds	
<b>Write</b>	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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe221	No	no	No
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int32	noprot	read salvo pause timeout in seconds (see SALVO_PAUSE_STATE)	
<b>Write</b>	t_int32	service (VJD only)	set salvo pause timeout in seconds, default value is 60s (see SALVO_PAUSE_STATE)	



# Keyboard and IP Matrix Operations

## INFO\_OVERLAY\_MODE

	Tag code	NumDes	Message	SNMP Support
	0xe219	no	no	no
	Datatype	Access Level	Description	
<b>Read</b>			not supported	
<b>Write</b>	t_int32	service, user (VJD only)	video panel info text overlay mode 0: off 1: video panel (coder) info 2: video source (camera) info	

## KBD\_CONNECT\_CAMERA

	Tag code	NumDes	Message	SNMP Support
	0xe211	yes	no	no
	Datatype	Access Level	Description	
<b>Read</b>	p_octet	service, user (VJD only)	get camera key (0xFFFF: monitor is not connected to a camera from the camera table)	
<b>Write</b>	p_octet	service, user (VJD only)	connect a camera from the camera table to a video panel (monitor)	

## Payload Structure

<b>Monitor Key</b> 2 Bytes	<b>Camera Key</b> 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	
<b>Read</b>	p_octet	service, user (VJD only)	get salvo key (0xFFFF: monitor is not connected to a salvo from the salvo table)	
<b>Write</b>	p_octet	service, user (VJD only)	connect a salvo from the salvo table to a video panel (monitor)	

## Payload Structure

<b>Monitor Key</b> 2 Bytes	<b>Salvo Key</b> 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	service, user (VJD only)	go-to preset

## Payload Structure

<b>Monitor Key</b> 2 Bytes	<b>Preset</b> 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 send 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 ist 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 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	
<b>Read</b>			not supported	
<b>Write</b>	p_octet	service, user (VJD only)	store preset	

### Payload Structure

<b>Monitor Key</b> 2 Bytes	<b>Preset</b> 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 send 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 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	
<b>Read</b>			not supported	
<b>Write</b>	p_octet	service, user (VJD only)	change the number of video panels	

### Payload Structure

<b>Line</b> 2 Bytes	<b>Layout Step Code</b> 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	
<b>Read</b>		not supported	
<b>Write</b>	p_octet	service, user	manipulate pan, tilt, zoom, iris and focus settings of a video panel

## Payload Structure

Line	Coder	Preset
1 Byte	1 Byte	2 Bytes
<b>Pan Speed</b>		
4 Bytes $([-1..1] * 32768 \text{ as int32})$		
<b>Tilt Speed</b>		
4 Bytes $([-1..1] * 32768 \text{ as int32})$		
<b>Zoom Speed</b>		
4 Bytes $([-1..1] * 32768 \text{ as int32})$		
<b>Iris Speed</b>		
4 Bytes $([-1..1] * 32768 \text{ as int32})$		
<b>Focus Speed</b>		
4 Bytes $([-1..1] * 32768 \text{ as int32})$		

### 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  $(\text{int32})(0.12 * 32768) = (\text{int32})3932,16 = 3932$

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 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**

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe21b	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>			not supported	
<b>Write</b>	p_octet	service, user	set fullscreen tile and mode	

**Payload Structure**

<b>Line</b>	<b>Coder</b>	<b>Reserved</b>	<b>Fullscreen Mode</b>
1 Byte	1 Byte	1 Byte	1 Byte

**Fullscreen Mode**

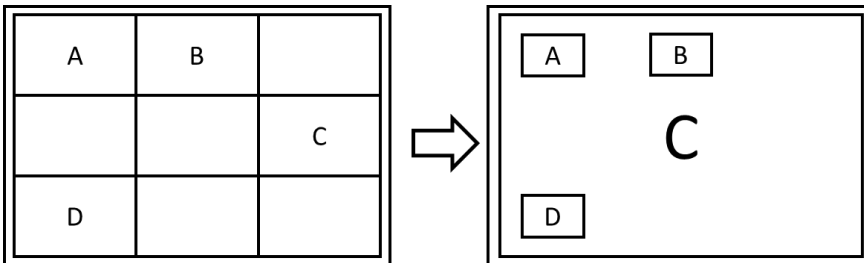
0: Off (coder is ignored)

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**

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

**Payload Structure**

<b>Line</b>	<b>Coder</b>	<b>Reserved</b>	<b>Freeze Flags</b>
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	
<b>Read</b>		not supported	
<b>Write</b>	t_octet	service, user (VJD only)	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	
<b>Read</b>		not supported	
<b>Write</b>	p_octet	service, user (VJD only)	Select highlighted video panel

#### Payload Structure Request

<b>Line</b> 1 Byte	<b>Coder</b> 1 Byte	<b>Reserved</b> 1 Byte	<b>Highlighting Index</b> 1 Byte
<b>Reserved</b> 4 Bytes			

#### Payload Structure Response

<b>Notification</b> 4 Bytes
<b>Reserved</b> 4 Bytes

#### Highlighting Index

A highlighted video panel is displayed with a different border colour. There are 4 different highlighting colours 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 (colours) 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 colour 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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe225	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>			not supported	
<b>Write</b>	t_int32	service, user (VJD only)	favorite index [0..9]	

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

Persistence of digital zoom setting is controlled by DIGITAL\_ZOOM\_PERSISTENCE\_MODE.

### LOAD\_FAVORITE

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe21c	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>			not supported	
<b>Write</b>	t_int32	service, user (VJD only)	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

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe21f	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int32	service, user	Reads the number of licensed channels of this decoder for IP matrix keyboard operations	
<b>Write</b>			not supported	

### KBD\_LICENSED\_CHANNELS\_GROUP

	<b>Tag code</b>	<b>NumDes</b>	<b>Message</b>	<b>SNMP Support</b>
	0xe223	no	no	no
	<b>Datatype</b>	<b>Access Level</b>	<b>Description</b>	
<b>Read</b>	t_int32	service, user	Reads the number of licensed channels across the whole decoder group for IP matrix keyboard operations	
<b>Write</b>			not supported	



### SERIAL\_PORT\_APP\_VAL

	Tag code	NumDes	Message	SNMP Support
	0x01f1	yes	no	no
	Datatype	Access Level	Description	
<b>Read</b>	t_octet	service, user	Get current serial port mode	
<b>Write</b>	t_octet	Service (VJD only)	Set serial port mode	

**Numerical descriptor:** Serial port number (ignored, internal serial port number is always 1)

**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	
<b>Read</b>	p_octet	service, user	See description	
<b>Write</b>	p_octet	service, 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. It 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.

**Numerical descriptor:** Serial port number (ignored, internal serial port number is always 1)

**Payload Structure Write**

	16		32
<b>Options</b> 1 Byte	<b>Reserved</b> 1 Byte	<b>Lease Time</b> 2 Bytes	
<b>Transparent Data</b> N Bytes			

Options: Ignored

Lease Time: Ignored

Transparent Data: Data to be sent to serial port 1

**Payload Structure Response (Read & Write)**

<b>Code</b> 1 Byte	<b>Reserved</b> 3 Byte
-----------------------	---------------------------

32

**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**

<b>Transparent Data</b> N Bytes
------------------------------------

Transparent Data: Data received at serial port 1

# Configuration Sealing

## CONFIG\_SEALING\_ENABLED

	Tag code	NumDes	Message	SNMP Support
	0x0c8a	no	no	no
	Datatype	Access Level	Description	
<b>Read</b>	t_octet	noprof	get enabled state of config sealing 0=sealing disabled; 1=sealing enabled; 2=seal broken	
<b>Write</b>	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	
<b>Read</b>	p_octet	noprof	return status of config seal	
<b>Write</b>	p_octet	service	read only	

## Payload Structure

<b>Status</b> 1 Byte	<b>Reserved</b> 3 Bytes
<b>SealSetTimestamp</b> 4 Bytes	
<b>SealRandom</b> 4 Bytes	
<b>SystemTimestamp</b> 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 a 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)