RCP Plus Reference

(RCP Version 3)

Specification

Firmware version	major	7
	minor	50
	build	79

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.

Table Of Contents

DROPBOX_AUTH_STATUS
DROPBOX_TOKEN_V2
DROPBOX_AUTH_V2
ACCOUNT_STATUS
ADV_USER_SETTINGS
ADV_USER_SETTINGS_LIST
ADV_USER_OPTIONS
USER_AUTH_MODE
MIN_TLS_VERSION
ENABLE_MASTERPWD
ACCESS_OPTION
MASTERPWD_CHALLENGE
MASTERPWD_CHALLENGE_STATE
CERTIFICATE
CERTIFICATE_OPTIONS
CERTIFICATE_LIST
CERTIFICATE_REQUEST
CERTIFICATE_REQUEST_OPTIONS
CERTIFICATE_REQUEST_PROGRESS
CERTIFICATE_USAGE
CERTIFICATE_USAGE_OPTIONS
STREAM_SECURITY_V2
STREAM_SECURITY_OPTIONS
VIDEO_OVER_SSL
WIFI_CONFIG_TOKEN
LOGIN_LIMITER_MESSAGE

STATUS

NBR_OF_TEMP_SENS
TEMP_SENS
REDUNDANT_POWER
POWER_SUPPLY
NBR_OF_FANS
FAN_SPEED
MINIMUM_FAN_SPEED
BOOT_STATE
UPLOAD_PROGRESS
HEATER_MODE
CHECK_DEFAULT_BUTTON_STATE
CHECK_POS_FB_STATE
CLOUD_COMMISSIONING_STATUS
AMBIENT_LIGHT_LEVEL
LOW_AMBIENT_LIGHT_THRESHOLD
NBR_OF_HUMIDITY_SENSORS
HUMIDITY_VALUE

USER_TASK

VIPROC_ID
RULE_ENGINE_ID
VIPROC_ONOFF
VIPROC_CUSTOM_PARAMETERS
VIPROC_FIRE_PARAMETERS
VIPROC_CUSTOM_PARAMETERS_TAGS

VIPROC_GLOBAL_PARAMETERS
VIPROC_GLOBAL_PARAMETERS_TAGS
VIPROC_DLL_NAME
VIPROC_DLL_NAME_LIST
VIPROC_SAVE_REFERENCE_IMAGE
VIPROC_REFERENCE_IMAGE_FILENAME
START_VIPROC_CONFIG_EDITING
STOP_VIPROC_CONFIG_EDITING
CONT_VIPROC_CONFIG_EDITING
VIPROC_SCENE
NUMBER_OF_VIPROC_CONFIGS
VIPROC_DEFAULT_LIST
VIPROC_DEFAULT
ACTIVE_VIPROC_CONFIG
LOADED_VIPROC_CONFIG
VIPROC_CONFIG_NAME
LIST_OF_VIPROC_SCENES
VIPROC_TAGGED_CONFIG
VIPROC_TAGGED_CONFIG_INTERNAL
VIPROC_ALARM
ALARM_OVERVIEW
AUPROC_ALARM
AUPROC_CONFIG
AUPROC_MELPEGEL
AUPROC_NAME
VIPROC_ALARM_MASK
VIPROC_CONFIG_CHANGE_IN_RECORDING
VIPROC_MODE
VIPROC_WEEKLY_SCHEDULE
VIPROC_HOLIDAYS_SCHEDULE
PTZ_CONTROLLER_AVAILABLE
ENABLE_VCA
VIPROC_ALARM_AGGREGATION_TIME
VIPROC_ALARM_EXT_AGGREGATION_TIME
VIPROC_MOTION_DEBOUNCE_TIME
VCA_TASK_RUNNING_STATE
VIPROC_VERSION
TRANSPARENT_DATA_OVER_IP
AUTO_TRACKER_TRACK_OBJECT
AUTO_TRACKER_TRACK_OBJECT_POS
MODE_AUTO_TRACKER
STATUS_AUTO_TRACKER
SENSITIVITY_OBJECT_BASED_VCA
VIPROC_SENSITIVE_AREA
REFERENCE_IMAGE_CHECK_INFO_MESSAGE
IVA_COUNTER_VALUES
BEST_FACE
WRITE_XML_TO_VCD
SCHEDULED_PTZ_PROFILES
SCHEDULED_PTZ_PROFILE
FIRE_ALARM
VCA_MSK_POLY
VCA_MSK_OPTIONS
VCA_MASK_DOME_PTZ_POS

DEBUG

SYSLOG_LOG_LEVEL
SYSLOG_PROTOCOL
SYSLOG_HOST_STR

SYSLOG_PORT
SYSLOG_MESSAGE
CONFIG_SEALING_ENABLED
CONFIG_SEALING_STATUS
CLOUD_WATCH_SETTINGS
SYSUPTIME
SD_CARD_STATUS

AUX

OEM_DEVICE_NAME
OEM_EXT_ID
OEM_DEVICE_DOMAIN

LOCATION

CAMERA_POSITION
CAMERA_ORIENTATION
CAMERA_POSITION_OPTIONS
CAMERA_SURROUNDING
CAMERA_SURROUNDING_OPTIONS
CAMERA_LOCATION_METADATA
CAMERA_CALIBRATION
COMPLETE_CALIBRATION_ELEMENT
SHIFT_CALIBRATION_ELEMENT
SENSOR_ORIENTATION
PREDEFINED_MOUNTING_LIST
GETPROFILE_ALGO_PRESET_NAME_LIST
COMMISSION_SETUP

AUX

DEFAULTS
FACTORY_DEFAULTS
BOARD_RESET
BOOT_DEFAULT_APP
VIDEO_ENC_PRIO
LED_BLINKING
JPEG
CLUSTER_GROUP_SETTING
CLUSTER_ID
APP_OPTION
APP_OPTION_EXT
APP_OPTION_UNIT_ID
APP_OPTION_SET
CPU_LOAD_IDLE
CPU_LOAD_CODER
CPU_LOAD_VCA
STORAGE_LOAD
CPU_LOAD
SYSTEM_LOAD
CPU_COUNT
SANITY_CHECK
ONVIF_STREAM_URI_EX
NIGHT_MODE_STATE
BLOCK_BASED_STORAGE
POE_GRANTED_POWER
INSTALLER_SEQUENCE
UPLOAD_HISTORY
STATIC_SERVER_URL
GB28181
PRIVACY_MODE
RUN_QR_READER
PTZ_AUTO_ROTATE_MODE
POWER_MONITOR_VALUES
POWER_MONITOR_NAMES
AUXILIARY_POWER
BLUETOOTH_CTRL
BLUETOOTH_SCAN
LLDP_REQUESTED_POWER_TOTAL
LLDP_REQUESTED_POWER_CAM
LLDP_ALLOCATED_POWER_TOTAL
LLDP_POWER_ADDER
BASE_POE_CLASS
SIGNALLED_POE_CLASS
STRATOCAST_ONOFF
STRATOCAST_REGISTER
STRATOCAST_STATE
NBR_OF_IMU
OBSOLETE

ENCODER

MPEG2

MPEG_AUDIO_SAMPLING_FREQ

MPEG4

MPEG4_CURRENT_PARAMS
MPEG4_CURRENT_PARAMS_TRANSCODER
MPEG4_CURRENT_PARAMS_REL_CODER
MPEG4_NAME
MPEG4_BANDWIDTH_KBPS
MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT
MPEG4_BANDWIDTH_KBPS_HARD_LIMIT
MPEG4_INTRA_FRAME_DISTANCE
MPEG4_FRAME_SKIP_RATIO
MPEG4_RESOLUTION
MPEG4_FIELD_MODE
MPEG4_DEFAULTS
MPEG4_I_FRAME_QUANT
MPEG4_P_FRAME_QUANT
MPEG4_PARAMS_MAX_NUM
MPEG4_AVC_I_FRAME_QUANT
MPEG4_AVC_P_FRAME_QUANT
MPEG4_AVC_P_FRAME_QUANT_MIN
MPEG4_AVC_DELTA_IPQUANT
MPEG4_AVC_QUANT_ADJ_REGION_1
MPEG4_AVC_QUANT_ADJ_REGION_2
MPEG4_AVC_DEBLOCKING_ENABLE
MPEG4_AVC_DEBLOCKING_ALPHA
MPEG4_AVC_DEBLOCKING_BETA
MPEG4_AVC_CHROMA_QUANT_OFF
MPEG4_AVC_CODING_MODE
MPEG4_AVC_GOP_STRUCTURE
GOP_STRUCTURE_OPTIONS

VIDEO_ENC_P_REF_LIST_SIZE
GET_VIDEO_ENC_P_REF_LIST_SIZE_LIMIT
MPEG4_AVC_CABAC
MPEG4_AVC_BITRATE_OPTIMIZATION_OPTIONS
ENC_DYN_SCENE_CTRL
ENC_DYN_SCENE_CTRL_OPTIONS
GET_ENC_DYN_SCENE_CTRL_INFO
MPEG4_AVC_BITRATE_OPTIMIZATION
CODER_SPECIFIC_ENC_PROFILES
VIDEO_BITRATE_AVERAGING_PERIOD
VIDEO_QUALITY
JPEG_BANDWIDTH_KBPS
JPEG_BANDWIDTH_KBPS_SOFT_LIMIT
JPEG_BANDWIDTH_KBPS_HARD_LIMIT
VID_ENCODER_ON
VIDEO_ENCODER_STATUS
VIDEO_ENCODER_STATUS_EXT
EXT_ENCODER_BITRATE_STATISTICS
VIDEO_ENCODER_ENCODED_BYTES
ENC_CURRENT_RESOLUTION
ENC_PROFILE_BASIC_PARAMS
ENC_PROFILE_PARAMS
ROI
ROI_OPTIONS
VIDEO_ENC_TEMP_QUANT_ADJ
DPTZ_DYNAMIC_CAPS
DPTZ_STATIC_CAPS
FLIP_MIRROR
VCA_SHAPES

STAMPING

STAMP
ENC_STAMPING_PROPERTIES
NAME_STAMP_VAL
TIME_STAMP_VAL
TIME_STAMP_RESOLUTION
ALARM_DISP_VAL
ALARM_STRING
INFO_STAMP_VAL
LOGO_STAMP_VAL
STAMP_ATTR_NAME
STAMP_ATTR_TIME
STAMP_ATTR_ALARM
STAMP_ATTR_INFO
STAMP_ATTR_LOGO
STAMP_WIDTH

VIDEO_CURRENT_PARAMS_CODNBR
MAX_NBR_OF_ENC_STREAMS
WATERMARK
CODER_VIDEO_OPERATION_MODE
CODER_VIDEO_OPERATION_OPTION
CODER_VIDEO_OPERATION_MODE_OPTIONS
VIDEO_H264_ENC_CONFIG
VIDEO_H264_ENC_CONFIG_DEFAULTS
VIDEO_H264_ENC_CONFIG_BULK
ENC_BASE_OPERATION_MODE_TYPE
VIDEO_H264_ENC_BASE_OPERATION_MODE
VID_H264_ENC_BASE_OPERATION_MODE_CAPS
VIDEO_H264_ENC_CURRENT_PROFILE
JPEG_STREAM_SETUP
JPEG_STREAM_SETUP_OPTIONS_VERBOSE
JPEG_STREAM_FRAME_RATES
ENC_PROFILE_RESOLUTION_OPTIONS
TCP_RATE_CONTROL
TCP_BANDWIDTH_CHECK
TCP_BANDWIDTH_CHECK_RESULT
SEI_ENABLE
PIC_INFO
IMAGE_STABILIZATION_MODE
TC_PLUGIN_PARAMS
STREAM_PRIORITY
NBR_OF_ENC_STREAMS
REQ_FAST_UPDATE
MPEG4_INTRA_FRAME_REQUEST

VIDEO

INPUT

CAMERA_LENS_CURVE
LOW_LIGHT_MIN_FPS
VID_IN_CONTRAST
VID_IN_SATURATION
VID_IN_BRIGHTNESS
VID_IN_SAMPLING_MODE
VID_IN_WHITE_BALANCE_MODE
VID_IN_RGB_GAIN
VID_IN_MIRROR
VIDEO_TERMINATION_RESISTOR_ON
VIDEO_TERMINATION_RESISTOR_OFF
VIDEO_INPUT_FORMAT
VIDEO_INPUT_FORMAT_EX
VIDEO_INPUT_FORMAT_EX_OPTIONS
VIDEO_INPUT_FORMAT_EX_VERBOSE
H264_ENC_BASE_OP_MODE_CAPS_VERBOSE
VIDEO_INPUT_DESCRIPTION
INPUT_SOURCE_VAL
NBR_OF_VIDEO_IN
PRIV_MSK
PRIV_MSK_POLY
PRIV_MASK_DOME_PTZ_POS
PRIV_MSK_OPTIONS
STATIC_PRIV_MSK_OVERLAY
STATIC_PRIV_MSK_OVERLAY_OPTIONS
VIDEO_STATIC_SCENE_REGIONS
VIN_BASE_FRAMERATE
ILLUMINATION_MODE
MAINS_FREQUENCY
VIRTUAL_LINES
VIDEO_LINE_DEWARPING_MODE
VIDEO_LINE_DEWARPING_MODE_CAPS
VIDEO_LINE_DUO

OUTPUT

VIDEO_OUT_STANDARD
VIDEO_OUT_STANDARD_FORCE
VIDEO_OUT_CROPPING
VIDEO_OUT_DISPLAY_FORMAT
MONITOR_NAME
VID_DECODER_ON
LOGO
DEC_SHOW_FREEZE
DECODER_MODE
DECODER_LAYOUT_LIST
DECODER_LAYOUT
VIDEO_OUT_STANDARD_SPEC
VIDEO_OUT_CURRENT_SPEC
VIDEO_OUT_MONITOR_SPEC
VIDEO_OUT_OVERSCAN
DECODED_FRAMES
NBR_OF_VIDEO_OUT
DECODER_DELAY
DECODING_ERROR

STAMPING

STAMP_ATTR_DEC_FREEZE
STAMP_DEC_FREEZE_STRING
STAMP_ATTR_NO_LINK
STAMP_NO_LINK_STRING

OSD

ENABLE_OSD
OSD_ACCESS
OSD_POS

AUDIO

AUDIO_INPUT_LEVEL
AUDIO_OUTPUT_LEVEL
AUDIO_ON_OFF
AUDIO_STARTUP_SOUND
AUDIO_NOTIFICATION_SOUND
AUDIO_INPUT
AUDIO_OUTPUT
AUDIO_INPUT_MAX
AUDIO_OUTPUT_MAX
AUDIO_MIC_LEVEL
AUDIO_MIC_MAX
AUDIO_LOUDSPEAKER_ON_OFF
AUDIO_OPTIONS
AUDIO_INPUT_PEEK
AUDIO_OUTPUT_PEEK
NBR_OF_AUDIO_OUT
NBR_OF_AUDIO_IN
AUDIO_AAC_BITRATE
VIRTUAL_AUDIO_LINES

EXTERNAL

CLIENTS

EXTERNAL_CLIENTS_LIST
EXTERNAL_CLIENT
EXTERNAL_CLIENT_EVENT

ALARM

SOURCE

ALARM_INPUT_LH_VAL
INPUT_PIN_NAME
ALARM_INPUT_SUPERVISED
ALARM_INPUT_CAPABILITIES
EVENT_SCRIPT
VCD_OPERATOR_PARAMS
VIPROC_RE_TASK_NAMES
VIPROC_DETECTOR_PARAMS
RAW_IMAGE
VCA_INPUT_FORMAT

CONNECTION

ALARM_CONNECT_TO_IP_STR
ALARM_CONNECT_TO_IP
NBR_OF_ALTERNATIVE_ALARM_IPS
AUTO_DISCONNECT_TIME
DEFAULT_CONNECTION_MODE
STD_MEDIA_CONNECTION_DIRECTION
STD_MEDIA_ENCAPSULATION_PROTOKOL
DEFAULT_CAM
ALARM_CONNECTION_DESTINATION_PORT
ALARM_CONNECTION_USE_SSL
CONNECT_URL

DIGITAL_IO

INPUT

VIDEO_ALARM_STATE
INPUT_PIN_STATE
MOTION_ALARM_STATE
RELAY_OUTPUT_STATE
RELAY_OUTPUT_MODE
NBR_OF_ALARM_IN
NBR_OF_ALARM_OUT
NBR_OF_MOTION_DETECTORS
HD_MGR_SIGNAL_ALARM
VIRTUAL_ALARM_STATE
SET_VIRTUAL_ALARM_ID
NBR_OF_VIRTUAL_ALARMS
MANIPULATION_ALARM_STATE
NBR_OF_MANIPULATION_ALARMS
VIRTUAL_ALARM_PARAMETER
PIR_SENSITIVITY
PIR_STATUS
NBR_OF_PIR
PIR_ALARM_STATE
ILLUMINATOR_OPTIONS
ILLUMINATOR_STATE
ILLUMINATION_INTENSITY
VCA_BRIGHTNESS_INFO
VCA_SUGGESTED_FIREMASK

OUTPUT

RELAIS_NAME
RELAIS_SWITCH

SERIAL

SERIAL_PORT_APP_VAL
SERIAL_PORT_RATE
SERIAL_PORT_BITS
SERIAL_PORT_STBITS
SERIAL_PORT_PAR
SERIAL_PORT_MODE_VAL
SERIAL_PORT_HD_MODE_VAL
SERIAL_PORT_HANDSHAKE
SERIAL_ACTS_ACCESS_RIGHTS
BICOM_COMMAND
BICOM_SRV_CONNECTED
BICOM_SUBCOMPONENTS_LIST

KBD_CONFIG_CAMERA
KBD_CONFIG_CAMERA_STR
KBD_CONFIG_MONITOR
KBD_CONFIG_MONITOR_STR
KBD_CONNECT_PARAMS
KBD_PASSWORD_CAMERA
KBD_PASSWORD
KBD_CONFIG_SALVO
KBD_KEY_LABEL
KBD_SET_ALARM
TPPP_PORT

NETWORK

INTERFACE

MAC_ADDRESS
IP
IP_STR
SUBNET
SUBNET_STR
GATEWAY_IP_V6_STRING
IP_V6_PREFIX_LEN
IP_V6_STR
STATELESS_IP_V6_PREFIX_LEN
STATELESS_IP_V6_STR
GATEWAY_IP_STR
DNS_SERVER_IP
DNS_SERVER_IP_STRING
APPLY_NETWORK_SETTINGS
ETH_LINK
ETH_LINK_STATUS
ETH_LINK_TROUGHPUT
ETH_TX_PKT_BURST
NBR_OF_EXT_ETH_PORTS
NBR_OF_EXT_ETH_COPPER_PORTS
NBR_OF_EXT_ETH_FIBER_PORTS
LED_CAPS
ENABLE_TRAFFIC_LED
SND_MSS
EAP_IDENTITY
EAP_GET_IDENTITY_LIST
EAP_ENABLE
EAP_PASSWORD
SWITCH_TRUNKING
SWITCH_LACP_ENABLE
SWITCH_LACP_KEY
SWITCH_RSTP_PRIO
SWITCH_RSTP_HELLO_TIME
SWITCH_RSTP_MAX_AGE
SWITCH_RSTP_FWD_DELAY
SWITCH_RSTP_VERSION
SWITCH_RSTP_PORT_ENABLE
SWITCH_RSTP_PORT_EDGE
SWITCH_RSTP_PORT_PATHCOST
BACKPLANE_TYPE
BACKPLANE_FW_VERSION
SWITCH_IGMP_IP
SWITCH_IGMP_SNOOPING_ENABLE
SWITCH_MAC_IS_SET
REBOOT_SWITCH
WLAN_SSID
WLAN_WPA_PSK
WLAN_IOCTRL
PORT_FC_MODE
WLAN_SCAN
WLAN_LINK_TEST
WLAN_OPERATING_MODE
SWITCH_POST_UPDATE_ACTION
WLAN_REGION_CODE
WLAN_LINK_QUALITY
WLAN_WPS_SETUP
WLAN_WPS_STATUS
WLAN_DEFAULT_CHANNEL
SOCKET_KNOCKER_MODE
SOCKET_KNOCKER_DESTINATION
SOCKET_KNOCKER_DESTINATION_NAME
SOCKET_KNOCKER_STATUS
FTP_SERVER_PORT
FTP_PWD
ACCOUNT_LIST
ACCOUNTS_CREATE_FOLDER
ACCOUNTS_DELETE_FOLDER
FTP_CWD

FTP_CDUP
START_CLIENT
STOP_CLIENT
FTP_FILE_NAME
ACCOUNT_LOGIN_TEST
DIFF_SERV_VAL
DIFF_SERV_POST_ALARM_TIME
IPV4_FILTER
UPLINK_KBPS
TC_CLIENT_PARAMS
OBEY_ICMP_REDICRECTS
NETWORK_SERVICES
CBS_COMMISSION

CBS_STATUS
CBS_DESTINATION

MULTICAST

MULTICAST_VIDEO_GROUP_IP_STR
MULTICAST_VIDEO_GROUP_IP
MULTICAST_AUDIO_GROUP_IP_STR
MULTICAST_AUDIO_GROUP_IP
MULTICAST_VIDEO_PORT_STR
MULTICAST_VIDEO_PORT
MULTICAST_AUDIO_PORT_STR
MULTICAST_AUDIO_PORT
MULTICAST_TTL
IGMP_VERSION
STREAMING_VAL
VIDEO_ENC_STREAMING
AUDIO_ENC_STREAMING
START_MULTICAST_STREAMING
AUDIO_STREAMING_ENCODING
JOIN_STREAM_NOW

DNS

DYNDNS_SERVER
DYNDNS_TIMEOUT
DYNDNS_SERVER_REPLY

DYNDNS

DYNDNS_HOST_NAME
DYNDNS_USER_NAME
DYNDNS_PASSWORD
DYNDNS_ENABLE
DYNDNS_STATE
DYNDNS_LAST_REGISTER
DYNDNS_FORCE_REGISTER_NOW
DYNDNS_PROVIDER
DYNDNS_STATUS_MAIL_ONOFF
DYNDNS_MAIL_SMTP_SRV_IP_STR
DYNDNS_MAIL_SMTP_LOGIN
DYNDNS_MAIL_SMTP_PASS
DYNDNS_MAIL_DEST_EMAIL_ADDR
DYNDNS_MAIL_SENDER_NAME
DYNDNS_MAIL_TEST_SEND

DISCOVERY

AUTODETECT_REPLY_GROUP
UNSOLICITED_AUTODETECT_REPLY_TIME
DISCOVER_PORT

HTTP

LOCAL_HTTP_PORT
LOCAL_HTTPS_PORT
RCP_SERVER_PORT
TELNET_PORT
RTSP_PORT
ENABLE_UPNP
ENABLE_ONVIF
GET_RTSP_SESSION_ID
HSTS_ENABLED
ALLOW_BASIC_HTTP_AUTH_ON_NON_SSL_SOCK
PROTECT_HTTP_COOKIE

GROUP DEVICES

TCP_FWD
PREPARE_FOR_RECORDING
ADD_REMOTE_DEVICE
UPNP_SEARCH_IP_CONN_SERVICE
UPNP_TCP_FWD

DHCP

DHCP_VAL
DHCP_ON
DHCP_OFF
DHCP_STABLE
DHCP_COMPLIANCY

SNMP

SNMP_SRV_PORT
SNMP_TRAPS_HOST
SNMP_TRAPS_HOST_STR
NBR_OF_TRAPS_HOSTS
SNMP_TRAP_LIST
PMPP_PORT
PMPP_ADDRESS
SYSCONTACT
SYSLOCATION
SNMP_READ_COMMUNITY
SNMP_WRITE_COMMUNITY
SNMP_TRAP_COMMUNITY
SNMP_SERVER_MODE
SNMP_USER_PROFILE

SG

STREAMING_GATEWAY_CONFIG
STREAMING_GATEWAY_ACTIVE_LINES
STREAMING_GATEWAY_MAX_LINES

ISCSI

ISCSI_IP
ISCSI_PORT
ISCSI_LUN
ISCSI_TARGET_IDX
ISCSI_TARGET
ISCSI_TCP_CONNECTIONS
ISCSI_DISCOVERY
ISCSI_TARGET_PWD
ISCSI_LOCK_OVERRIDE
ISCSI_LOCK_RELEASE_ON_LEAVE
ISCSI_INITIATOR_NAME
ISCSI_INITIATOR_NAME_EXTENTION
ISCSI_SERVER_STATE
ISCSI_MNI
ISCSI_AUTH
ISCSI_SEG_SIZE
ISCSI_DATARATE

ISCSI_LOWERDATARATE
ISCSI_READDATARATE
ISCSI_MULTIPATH
ISCSI_MULTIPATH_STATE
ISCSI_FLUSH_DISVCACHE
ISCSI_DISV_CACHE
ISCSI_MP_DISCOVER

SMB

STORAGE

RECORDING

HD_MAX_NUMBER_OF_PARTITIONS
HD_MAX_SLICES_PER_TRACK
HD_MAX_ALARM_TRACKS_PER_PARTITION
HD_PARTITIONS_RECORDING
HD_PARTITION_RECORDING
HD_PARTITION_RECORDING_SECONDARY
HD_PARTITION_FILE_INFO
HD_PARTITION_PROP
HD_PARTITION_PROP_SECONDARY
HD_SIZE_MB
HD_PARTITION_GEO
HD_RECORD_SCHEDULE
HD_RECORD_SCHEDULE_SECONDARY
HD_RECORD_HOLIDAYS
HD_RECORD_PROFILES
HD_RECORD_PROFILES_SECONDARY
HD_RECORD_PROFILES_V2
HD_RECORD_PROFILES_V2_SECONDARY
HD_ALARM_MOTION
HD_ALARM_INPUT
HD_MGR_START
HD_MGR_START_SECONDARY
HD_MGR_STOP
HD_MGR_STOP_SECONDARY
HD_MPEG4_ACTIVE
HD_RECORDING_REPORT
HD_RECORDING_REPORT_SECONDARY
RECORDING_STATUS
RECORDING_RETENTION_TIME
RECORDING_RETENTION_TIME_SECONDARY
MAX_RECORDING_RETENTION_TIME
REMOTE_REC_DEVICE
REC_MGNT
HD_MGR_REC_STATUS
HD_MGR_REC_STATUS_SECONDARY
HD_FILE_INFO
HD_FILE_INFO_SECONDARY
HD_RELOAD_PARTITION_FILE_INFO
HD_RECORDING_ACTIVE
HD_REPLAY_ENCRYPTED_DATA
HD_REPLAY_AUTHENTICITY_OUT
START_RECORD
SET_REC_BUFFER_SIZE
AUDIO_REC_FORMAT
MANAGING_VRM
REC_STORAGE_REQ_CFG
RECORDING_BUFFER_LEVEL
HD_REC_BUFFER
MAX_GOP_LENGTH_VALUE
ALARM_BACKUP_REC_SPEED_LIMIT
BACKUP_RECORDING_STATUS
BACKUP_RECORDING_STATUS_MSG_THRESHOLD
BUFFERED_RECORDING_MODE
WPS_BUTTON_ENABLED
HD_FORMAT_STORAGE_ERROR_LOG_CNT
SD_CARD_LIFE_SPAN_STATUS
SD_CARD_LIFE_SPAN_ALARM_THRESHOLD
SD_CARD_LIFE_SPAN_MANF
SD_CARD_LIFE_SPAN_ENABLE
NBR_OF_SD_CARD_SLOTS
EXT_RECORDER_BITRATE_STATISTICS

REPLAY

HD_REPLAY_START
HD_REPLAY_START_EX
HD_REPLAY_STOP
HD_REPLAY_STOP_TIME
HD_REPLAY_SEEK_TIME
HD_REPLAY_SEEK_IFRAME
HD_REPLAY_EVENT_INFO
HD_REPLAY_PARTITION_EVENT_INFO
HD_REPLAY_MOTION_SAMPLES
HD_REPLAY_FAST_INTRA_DELAY
HD_REPLAY_FAST_INTRA_FPS
HD_REPLAY_LIVE
HD_REPLAY_SIZE_INFO
HD_REPLAY_VCD_LAYER
HD_REPLAY_VCD_CONFIG_ID
HD_REPLAY_FORENSIC_SEARCH_SETUP
HD_REPLAY_FORENSIC_SEARCH_CANCEL
HD_REPLAY_FORENSIC_SEARCH_RESULT
HD_REPLAY_PREFETCH_JPEGS
HD_REPLAY_CUSTOM_SETTINGS
HD_REPLAY_CERTIFICATES_LIST
HD_REPLAY_SEQUENCE_VERIFY
BACKUP
BACKUP_MAX_KBPS
BACKUP_STATUS
BACKUP_STOP

FAT32

FORMAT_FS
FORMAT_FS_STATUS

SPAN

FORMAT_FS_SPAN
FORMAT_FS_SPANS_STATUS
SPAN_ADDRESS_LIST
SPAN_ADDRESS_LIST_NEW
SPAN_HDR_ACCESS
START_SPAN_RECORD
SPAN_PARTITION_PROP

Transport Protocol

The transport protocol for this remote control must be TCP. All VideoJets will establish a TCP listen socket on port **1756**. Any remote control must be sent to this port. Multiple RCP connections from the same endpoints are allowed. The maximum number of RCP connections on a single endpoint is limited to 50 connections at the same time. Alternatively, a connection can be made using a HTTP tunnel. See this chapter for details.

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.

<table>
<tr><td colspan="4">16</td><td colspan="3">32</td></tr>
<tr><td colspan="4">Tag
2 Bytes</td><td>Data Type
1 Byte</td><td>Version
4 Bits</td><td>R/W
4 Bits</td></tr>
<tr><td>C
1</td><td>T
1</td><td>Action
6 Bits</td><td>Reserved
1 Byte</td><td colspan="3">Client ID
2 Bytes</td></tr>
<tr><td colspan="7">Session ID
4 Bytes</td></tr>
<tr><td colspan="4">Numeric Descriptor
2 Bytes</td><td colspan="3">Payload Length
2 Bytes</td></tr>
<tr><td colspan="4">8</td><td colspan="3">24</td></tr>
</table>

Tag

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

Data Type

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

Values:

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

Version

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

R/W

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

Values:

Read	0x0
Write	0x1

T

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 .

Packetizing in the TCP Stream

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

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

Version

TPKT version 3.

Reserved

Should be set to zero.

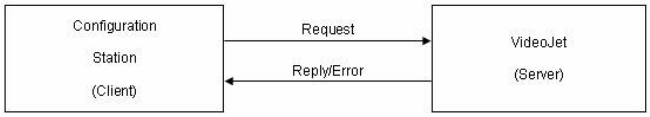
Length

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

RCP Protocol Procedure

Request / Reply

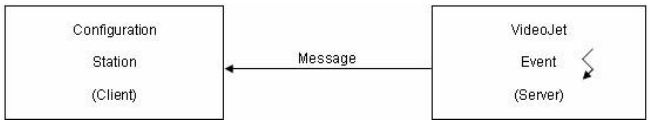
The VideoJet unit will send back a reply on each incoming request. In some protocol functions the reply will carry no data.



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

Messages

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



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

RCP Errors

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

RCP_ERROR_UNKNOWN	0xFF
RCP_ERROR_INVALID_VERSION	0x10
RCP_ERROR_NOT_REGISTERED	0x20
RCP_ERROR_INVALID_CLIENT_ID	0x21
RCP_ERROR_INVALID_METHOD	0x30
RCP_ERROR_INVALID_CMD	0x40
RCP_ERROR_INVALID_ACCESS_TYPE	0x50
RCP_ERROR_INVALID_DATA_TYPE	0x60
RCP_ERROR_WRITE_ERROR	0x70
RCP_ERROR_PACKET_SIZE	0x80
RCP_ERROR_READ_NOT_SUPPORTED	0x90
RCP_ERROR_INVALID_AUTH_LEVEL	0xa0
RCP_ERROR_INVAILD_SESSION_ID	0xb0
RCP_ERROR_TRY_LATER	0xc0
RCP_ERROR_TIMEOUT	0xd0
RCP_ERROR_NO_LICENCE (used by NVR only)	0xe0
RCP_ERROR_COMMAND_SPECIFIC	0xf0
RCP_ERROR_ADDRESS_FORMAT	0xf1

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

A list of all defined command specific error codes can be found in the Appendix.

RCP over a HTTP tunnel

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

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

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

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

Example:

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

Receiving media data

The VideoJet is capable for tunneling media data through HTTP. In some network environments this might be helpful.

When using this mechanism, the connect primitive must set the "RTP over TCP" for the media encapsulation protocol. The response for Media host and port can be ignored. After that, a TCP connection to the HTTP port must be established. By sending the string

GET /media_tunnel/SSSSSSSS/CC/DD/LL/CO HTTP/1.0\r\n\r\n

SSSSSSS	The session ID returned from the connect primitive (padded with leading zeros to 8)
CC	The media type (see connect primitive; 01=video, 02=audio)
DD	direction of media (01=receive media, 00=transmit media)
LL	the line input number
CO	The relative coder number

When the HTTP server inside the VideoJet received this string, it passes this socket to the RCP. From that point on, media data will be transferred over this socket using TPKT encapsulation (see chapter Packetizing in the TCP Stream).

The coder numbers that have to be used to establish the media tunnels is a relative one that isn't unique when using video as media type. The video and meta data can have the same relative coder number and have the same media type. In that case the optional index parameter might be used to distinguish the media channels:

GET /media_tunnel/SSSSSSSS/CC/DD/LL/CO?index=n HTTP/1.0\r\n\r\n

where n is the number of the initial connect primitive channel position starting at 1.

E.g the ConnectPrimitive request contains a Get Video, a Get Audio and a Get Meta section, then the index for video would be 1, the index for audio 2 and the index for meta would be 3.

When all sockets of a session are in data sending mode, a RCP message MEDIA_SOCKETS_COMPLETE with the associated session ID will be sent out to the initiator of the session. This is useful when a session is used for replay; the replay PLAY command can securely be applied when the MEDIA_SOCKETS_COMPLETE message has been arrived.

RCP over CGI

RCP+ command my also be encapsulated in CGI (Common Gateway Interface) using a standard web browsers URL. All needed parameters are passed to the HTTP server of the video server which tunnels the RCP+ commands to the RCP server. The reply returned is a text based XML structure.

The CGI interface may also be used by a HTTP client written from the scratch based on RFC 1945 (HTTP 1.0) using 'GET' method. The unit's HTTP server will only examine 'Connection: Keep-Alive' and optional 'Authorization: ' header fields. See RFC for implementing details.

Valid CGI parameters:

command

Values:

The RCP+ command tag number

type

Values:

F_FLAG

T_OCTET

T_WORD

T_DWORD

P_OCTET

P_STRING

P_UNICODE

direction

Values:

READ

WRITE

num

Values:

The numeric parameter

payload

Values:

The payload as readable string for payload type P_STRING

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

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

Optional CGI parameters:

idstring

Values:

User defined parameter which will not be processed by the unit; will return unchanged in the reply

sessionid

Values:

Context specific session ID for this command

Multiple parameters are passed in CGI manner with '&' in between two parameter-value pairs.

A sample request to readout the units hardware version will look like

http://ip.ip.ip.ip/rcp.xml?command=0x002e&type=P_STRING&direction=READ

The reply XML structure will then have this format:

```
<rcp>
   <command>
      <hex>0x002e</hex>
      <dec>46</dec>
   </command>
   <type>T_DWORD</type>
   <direction>READ</direction>
   <num>0</num>
   <idstring />
   <payload />
    <result>
      <hex>0xf0000f43</hex>
      <dec>4026535747</dec>
   </result>
</rcp>
```

The upper part contains the request, the <result> contains the reply. Numerical values are presented in <hex> and <dec>, strings, unicodes and p_octet in <str> sections.

Message handling

RCP+ messages will be processed and received by the CGI client using a poll mechanism. A CGI command set with the requested messages command tag numbers need to be issued. After issuing a request, the reply returns immediately after a message has been sent or the default timeout of 1000 ms has been expired. When the timeout expires, an empty message list is returned. Otherwise the received messages are listed. If there are several messages in the list, they are ordered from oldest to newest. The default timeout can be altered by setting the 'collectms' CGI value to the appropriate number of milliseconds. In order to ensure correct assignment of messages in case several clients are polling for messages or if a polling client uses different socket connections, a unique message domain ID (CGI parameter 'msgdomainID') should be provided. Messages are then collected separately for each ID and can be uniquely assigned to the polling client even if requesting via different socket connections.

Valid CGI commands to control messages:

message

Values:

One or a list of requested message RCP+ command tag numbers separated by '$'

collectms

Values:

Time in milliseconds to collect messages before returning with 'No messages'; default is 1000 ms; maximum is 60000 (1 minute)

msgdomainID

Values:

Unique ID (maximum 4 Bytes) to identify a certain polling client and to e.g. ensure correct message assignment to clients sending requests via different socket connections. The ID is choosen by the polling client

A buffer mechanism ensures that no messages will be lost during two consecutive poll cycles. The maximum buffering time is 10 seconds after the last poll request returned (either with a message list or due to timeout).

A sample request to receive all connection related messages (CONF_CONNECT_TO) will look like:

http://ip.ip.ip.ip/rcp.xml?message=0xffcc&collectms=5000

The reply XML structure will then have this format:

```
<message_list>
  <msg>
    <command>0xffcc</command>
    <num>0</num>
    <sessionID>0x7063001f</sessionID>
    <hex>0xa00a00340000000001010000010100001</hex>
  </msg>
  <msg>
    <command>0xffcc</command>
    <num>0</num>
    <sessionID>0x70630020</sessionID>
    <hex>0xa00a00340000000001010000010100001</hex>
  </msg>
</message_list>
```

Authentication

The RCP+ server needs proper authentication to process protected commands. The CGI interface provides two basic authentication schemes to be used.

Using http authentication

In this case, HTTP header authentication (basic or digest) must be present in the request. The internal HTTP server will pass the granted authorization level to the RCP+ server.

Using session cookie

For each successful HTTP authentication (for all available resources like html pages, images...) a session cookie will be returned by the HTTP server. When this cookie is present in any further HTTP connections, the same authorization level will be granted as for the originating connection. The session cookie will remain active as long as at least one HTTP connection remains open. In this case the internal HTTP server will pass the granted authorization level to the RCP+ server.

IPv6

For RCP commands which are defined as T_DWORD or P_OCTET and contain DWORD IP-Addresses
There is a extension to read or write also IPv6 addresses or Hostnames
e.g. URL:

http://ip.ip.ip.ip/rcp.xml?command=0x0a1f&type=T_DWORD&direction=WRITE&payload=0x00000000&ip-item0=fd00::55:77

or

http://ip.ip.ip.ip/rcp.xml?command=0x0a1f&type=T_DWORD&direction=WRITE&payload=0x00000000&ip-

item0=MyHostname.de

Where payload must contain a 4 byte index to the trailing list of ip-item0..ip-itemn

The reply XML structure will then have this format:

```
<rcp>
  <command>
    <hex>0x0a1f</hex>
    <dec>2591</dec>
  </command>
  <type>T_DWORD</type>
  <direction>READ</direction>
  <num>0</num>
  <idstring />
  <payload />
   <result>
    <hex>0x00000000</hex>
    <dec>0</dec>
    <ip-table>
      <item>
        <idx>0</idx>
        <type>IPV6</type>
        <str>fd00::55:77</str>
      </item>
    </ip-table>
  </result>
</rcp>
```

Where payload contains a 4 byte index to the ip-table list for each address contained in the command

if the command datatype is P_OCTET and the command does contain more IP-Addresses then the result will contain multipe item blocks.
Alternative formats for the item blocks are also possible:

```
      <item>
        <idx>1</idx>
        <type>IPV4</type>
        <str>192.168.1.2</str>
      </item>

      <item>
        <idx>2</idx>
        <type>NAME</type>
        <str>MyHostname.de</str>
      </item>
```

If a list is supplied the ALL addresses MUST be in the list also in the case that they are IPv4 type

Autodetecting Devices

Bosch Security Systems products are being detected on a network using an IP broadcast to either the static UDP port **1757** or to the configurable UDP discover port. The default port number for the configurable discover port is **1800**, and it can be modified with the RCP command CONF_DISCOVER_PORT (0x0976). In addition to the possibility to send a scan request packet to the broadcast destination address of 255.255.255.255, the discover port of the devices listens for scan requests sent to the configurable multicast group, which can be set by means of the RCP command CONF_AUTODETECT_REPLY_GROUP (0x0956). The default multicast detect group is 225.86.67.83.

Devices generate and transmit their replies to an autodetect request with a random delay of up to 2 seconds. When implementing your own detection software, please bear in mind that, when calculating the necessary waiting period until all replies are received, additionally to the possible span of max 2 seconds, you need to consider the time necessary to process the reply as well as the time the packet spends on the wire and in propagation devices like switches or routers on its way to and back from the device to be detected.

The scanned devices send their replies either to the source IP address of the request, or to the broadcast address 255.255.255.255. In case the source IP address of the request resides inside the same subnet, that is defined by the scanned device's IP address and it's subnet mask, the reply is addressed to the source IP address in the request. If the originator is not located in the same subnet as the scanned device, the reply is addressed to the broadcast address 255.255.255.255. In case of a scan request sent to the device via the multicast scan group, the reply is addressed to the same multicast group.

An autodetect scan request packet must consist of an autodetect request header, that may or may not be preceeded by an RCP plus protocol header for command id CONF_RCP_AUTODETECT (0xffdc). In case of an existing RCP plus protocol header in the scan request, the length field of the RCP plus protocol header must contain the length of the autodetect request header.

The format for the autodetect request header as well as for the autodetect reply is shown below:

Autodetect Request Header

16 32

0x99 1 Byte	**0x39** 1 Byte	**0xA4** 1 Byte	**0x27** 1 Byte
Sequence Number 4 Bytes			
Device Group 1 Byte	**Hardware ID** 1 Byte	**Reply Port** 2 Bytes	

8 24

Sequence Number

This sequence number must be a randomized number.

Device Group

Set the Device Group byte to 0xFF, to scan for all devices. Use value of 0x00 to scan for transmitters and 0x01 to scan for receivers.

Hardware ID

Set the Hardware ID byte to 0x00, to scan for all devices. Use the respective Hardware ID byte to scan for devices of that model.

Reply Port

Specifies the reply UDP port e.g. RP1 0x06; RP2 0xDE -> reply port: 1758

Additional information on hardware versions

32-Bit Hardware version Former Scheme:

16 32

ID 4 Bits	**R** 2 Bits	**FL** 2 Bits	**Serial Number** 2 Bytes	**Net** 4 Bits	**opt** 4 Bits

8 24

ID

Values:

NCID_NVR	0x1
NCID_VJ8000	0x2
NCID_VIP10	0x3
NCID_VIP1000	0x4
NCID_VJ400	0x6
NCID_VIP100	0x7
NCID_VJEX	0x8
NCID_VJ1000	0x9
NCID_VJ100	0xA
NCID_VJ10	0xB
NCID_VJ8008	0xC

NCID_VJ8004	0xD (latest unit carrying the old numbering scheme)
EscapeCode	0xF (used to switch over to new detection)

R

Reserved

FL

Values:

HARDWARE_TYPE_VIN	0x0
HARDWARE_TYPE_VOUT	0x1
HARDWARE_TYPE_VIN_OUT	0x2

Net

Values:

Bit1	HARDWARE_SYSTEM_OPT_ISDN
Bit3	HARDWARE_SYSTEM_ETHERNET

opt

Values:

Bit1	HARDWARE_OPT_AUDIO
Bit2	HARDWARE_OPT_HDD

32-Bit Hardware version New Scheme:

Esc 4 Bits	**R** 1	**M** 1	**FL** 2 Bits	**Reserved** 8 Bits	**ID** 1 Byte	**Net** 4 Bits	**Opt** 4 Bits

8 16 24 32

Esc

Must be set to 0xF

R

Reserved

M

Values:

Master	0
Slave	1

FL

Flavour

Values:

HARDWARE_TYPE_VIN	0x0
HARDWARE_TYPE_VOUT	0x1
HARDWARE_TYPE_VIN_OUT	0x2

ID

Values:

HARDWARE_ID_VIPX1	0x01
HARDWARE_ID_VIPX2	0x02
HARDWARE_ID_VIPXDEC	0x03
HARDWARE_ID_VJ_X10	0x05
HARDWARE_ID_VJ_X20	0x06
HARDWARE_ID_VJ_X40	0x07
HARDWARE_ID_VJ_X40_ECO	0x08
HARDWARE_ID_VJ_X10_ECO	0x09
HARDWARE_ID_VJ_X20_ECO	0x0A
HARDWARE_ID_IP_PANEL	0x0d
HARDWARE_ID_GEN4	0x0e
HARDWARE_ID_M1600	0x0f
HARDWARE_ID_FLEXIDOME	0x11
HARDWARE_ID_M1600_DEC	0x13
HARDWARE_ID_M1600_XFM	0x15
HARDWARE_ID_AUTODOME	0x16
HARDWARE_ID_NBC_225P	0x1a
HARDWARE_ID_VIPX1_XF	0x1e
HARDWARE_ID_NBC_225W	0x1f
HARDWARE_ID_NBC_255P	0x20
HARDWARE_ID_NBC_255W	0x21
HARDWARE_ID_AUTODOME_EASY_II	0x23

HARDWARE_ID_AUTODOME_EASY_II_E	0x24
HARDWARE_ID_VIPX1_XF_E	0x25
HARDWARE_ID_VJT_X20XF_E_2CH	0x26
HARDWARE_ID_VJT_X20XF_E_4CH	0x27
HARDWARE_ID_VIPX1_XF_W	0x28
HARDWARE_ID_VG5_AUTODOME_700	0x29
HARDWARE_ID_NDC_455_P	0x2a
HARDWARE_ID_NDC_455_P_IVA	0x2b
HARDWARE_ID_NBC_455_P	0x2c
HARDWARE_ID_NBC_455_P_IVA	0x2d
HARDWARE_ID_VG4_AUTODOME	0x2e
HARDWARE_ID_NDC_225_P	0x2f
HARDWARE_ID_NDC_255_P	0x30
HARDWARE_ID_VOT	0x32
HARDWARE_ID_NDC_274_P	0x35
HARDWARE_ID_NDC_284_P	0x36
HARDWARE_ID_NTC_265_PI	0x37
HARDWARE_ID_NDC_265_PIO	0x38
HARDWARE_ID_DINION_720P	0x39
HARDWARE_ID_NDN_822	0x3a
HARDWARE_ID_FLEXIDOME_720P	0x3b
HARDWARE_ID_NDC_265_W	0x3c
HARDWARE_ID_NDC_265_P	0x3d
HARDWARE_ID_NBC_265_P	0x3e
HARDWARE_ID_NDC_225_PI	0x3f
HARDWARE_ID_NTC_255_PI	0x40
HARDWARE_ID_JR_DOME_HD	0x41
HARDWARE_ID_JR_DOME_HD_FIXED	0x42
HARDWARE_ID_EX30_IR	0x43
HARDWARE_ID_GEN5_HD_PC	0x45
HARDWARE_ID_EX65	0x46
HARDWARE_ID_DINION_1080P	0x47
HARDWARE_ID_FLEXIDOME_1080P	0x48
HARDWARE_ID_HD_DECODER	0x49
HARDWARE_ID_GEN5_HD	0x4a
HARDWARE_ID_NER_L2	0x4b
HARDWARE_ID_VIP_MIC	0x4c
HARDWARE_ID_GEN5_A5_800	0x4d
HARDWARE_ID_NEVADA_TRANSCODER	0x4e
HARDWARE_ID_TESLA_BOXED	0x4f
HARDWARE_ID_TESLA_DOME	0x50
HARDWARE_ID_GEN5_A5_700	0x52
HARDWARE_ID_VJ_GENERIC_TRANSCODER	0x53
HARDWARE_ID_HD_DECODER_M	0x54
HARDWARE_ID_OASIS	0x55
HARDWARE_ID_GALILEO_BOXED	0x56
HARDWARE_ID_GALILEO_DOME	0x57
HARDWARE_ID_HUYGENS_KEPPLER	0x58
HARDWARE_ID_TESLA_KEPPLER	0x59
HARDWARE_ID_GALILEO_KEPPLER	0x5a
HARDWARE_ID_NUC_20002	0x5b
HARDWARE_ID_NUC_20012	0x5c
HARDWARE_ID_NUC_50022	0x5d
HARDWARE_ID_NUC_50051	0x5e
HARDWARE_ID_NPC_20002	0x5f
HARDWARE_ID_NPC_20012	0x60
HARDWARE_ID_NPC_20012_W	0x61
HARDWARE_ID_NIN_50022	0x62
HARDWARE_ID_NII_50022	0x63
HARDWARE_ID_NDN_50022	0x64
HARDWARE_ID_NDI_50022	0x65
HARDWARE_ID_NTI_50022	0x66
HARDWARE_ID_NIN_50051	0x67
HARDWARE_ID_NDN_50051	0x68
HARDWARE_ID_NAI_90022	0x69
HARDWARE_ID_NCN_90022	0x6a
HARDWARE_ID_NEVADA_DECODER	0x6c
HARDWARE_ID_NBN_50022_C	0x6d
HARDWARE_ID_NBN_40012_C	0x6e
HARDWARE_ID_NBN_80052	0x71
HARDWARE_ID_MIC_7000	0x72
HARDWARE_ID_EX65_HD	0x73
HARDWARE_ID_NBN_80122	0x74
HARDWARE_ID_NPC_20012_L	0x75

HARDWARE_ID_MIC_NPS	0x76
HARDWARE_ID_NBN_50051_C	0x78
HARDWARE_ID_NIN_40012	0x79
HARDWARE_ID_NII_40012	0x7a
HARDWARE_ID_NDN_40012	0x7b
HARDWARE_ID_NDI_40012	0x7c
HARDWARE_ID_NTI_40012	0x7d
HARDWARE_ID_NII_50051	0x7e
HARDWARE_ID_NDI_50051	0x7f
HARDWARE_ID_NEZ_4000	0x80
HARDWARE_ID_VEGA_3000_HD	0x81
HARDWARE_ID_VEGA_4000_HD	0x82
HARDWARE_ID_VEGA_5000_HD	0x83
HARDWARE_ID_VEGA_5000_MP	0x84
HARDWARE_ID_NIN_70122_180	0x85
HARDWARE_ID_NIN_70122_360	0x86
HARDWARE_ID_NEZ_5000	0x88
HARDWARE_ID_NEZ_5000_IR	0x89
HARDWARE_ID_ROLA	0x8b
HARDWARE_ID_NBN_80122_CA	0x8c
HARDWARE_ID_ISCSI_TARGET	0xf6
HARDWARE_ID_VRM_PROXY_16	0xf7
HARDWARE_ID_VRM_UL_APP	0xf8
HARDWARE_ID_VRM_LE_APP	0xf9
HARDWARE_ID_CAMNETWORK	0xfa
HARDWARE_ID_VRM_PROXY	0xfb
HARDWARE_ID_VRM	0xfc
HARDWARE_ID_VIDOS_SERVER	0xfd
HARDWARE_ID_VIDOS_MONITOR	0xfe

Net

Values:

Bit1	HARDWARE_SYSTEM_OPT_ISDN
Bit2	reserved
Bit3	HARDWARE_SYSTEM_ETHERNET
Bit4	reserved

opt

Values:

Bit1	HARDWARE_OPT_AUDIO
Bit2	HARDWARE_OPT_HDD
Bit3	reserved
Bit4	reserved

1st Reply Packet

0x99 1 Byte	**0x39** 1 Byte	**0xa4** 1 Byte	**0x27** 1 Byte
Sequence Number 4 Bytes			
Hardware Address 6 Bytes ...			
		0x03 1 Byte	**ID** 1 Byte
Device IP 4 Bytes			
Subnet Mask 4 Bytes			
Gatway IP 4 Bytes			
Flavor 1 Byte	**Connections** 1 Byte	**Reserved** 1 Bytes	**ID new** 1 Bytes

Sequence Number

Returns the sequence number from the request packet.

Hardware Address

MAC address of the device.

ID

Unit Hardware ID returned from request (will be 0xff when the new version layout is used).

Device IP

Current IP address of the device.

Subnet Mask

Current Subnet mask of the device.

Gatway IP

Current Gateway of the device.

Flavor

Flavor of this device.

Connections

Number of active connections.

ID new

The ID filed of the new version layout.

2nd Reply Packet

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="OEMExtXX:device-1-0">
<device>
   <deviceType>OEMExtXX:device:deviceIDYY</deviceType>
   <App>X</App>
   <friendlyName>VideoJet</friendlyName>
   <unitName>UnitName</unitName>
   <deviceGUID>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</deviceGUID>
   <serialNumber>XXXXXXXX:YYYYYYYY</serialNumber>
   <physAddress>XX-XX-XX-XX-XX-XX</physAddress>
   <sessionKey>XXXXXXXXXXXXXXXX</sessionKey>
   <unitIPAddress>XXX.XXX.XXX.XXX</unitIPAddress>
   <unitSubnetMask>XXX.XXX.XXX.XXX</unitSubnetMask>
   <unitGatewayIp>XXX.XXX.XXX.XXX</unitGatewayIp>
   <unitIPv6Address>XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX</unitIPv6Address>
   <activeConnections>XX</activeConnections>
   <RCPPort>XXXX</RCPPort>
   <HTTPPort>XXXX</HTTPPort>
   <HTTPSPort>XXXX</HTTPSPort>
   <dhcp>X</dhcp>
   <dhcpState>X</dhcpState>
   <cameraFrontendID>XXXX</cameraFrontendID>
   <Flags>X</Flags>
   <Lines>X</Lines>
   <BcJpeg>X</BcJpeg>
   <unitStatelessIPv6Address>XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX</unitStatelessIPv6Address>
   <scanID>X</scanID>
   <privacyMode>X</privacyMode>
   <ClusterMember>XXX.XXX.XXX.XXX</ClusterMember>
   <ClusterMemberMAC>XX-XX-XX-XX-XX-XX</ClusterMemberMAC>
   <ClusterID>X</ClusterID>
   <deviceCapabilities>XXXXXXXXXX...XX</deviceCapabilities>
</device>
</root>

<root xmlns="OEMExtXX:device-1-0">
XX is the OEM Extension; 0x00 for Bosch Security Systems; OEM specific

<deviceType>OEMExtXX:device:deviceIDYY</deviceType>
XX is the OEM Extension; 0x00 for Bosch Security Systems; OEM specific
YY is the VideoJet ID from the request packet

<App>X</App>
X is the application type (0: default application; 1: standard application)

<friendlyName>VideoJet</friendlyName>
The name and type of the VideoJet; type and OEM specific

<unitName>UnitName</unitName>
The name of the unit coded in UTF-8

<serialNumber>XXXXXXXX:YYYYYYYY</serialNumber>
XXXXXXX is the Hardware version of the VideoJet
YYYYYYY is the Softwareware version of the VideoJet

<physAddress> XX-XX-XX-XX-XX-XX </physAddress>
XX-XX-XX-XX-XX-XX is the MAC address of the VideoJet

<sessionKey>XXXXXXXXXXXXXXXX</sessionKey>
No common use

<unitIPAddress> XXX.XXX.XXX.XXX </unitIPAddress>
XXX.XXX.XXX.XXX is the current IP address of the VideoJet

<unitSubnetMask>XXX.XXX.XXX.XXX</unitSubnetMask>
XXX.XXX.XXX.XXX is the subnet mask of the VideoJet

<unitGatewayIp>XXX.XXX.XXX.XXX</unitGatewayIp>
XXX.XXX.XXX.XXX is the current gateway IP address of the VideoJet

<unitIPv6Address>XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX</unitIPv6Address>
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX is the current IPv6 address of the VideoJet e.g. FD00::1234:5678

<activeConnections>XX</activeConnections>
XX is the number of active media connections of the VideoJet

<RCPPort>XXXX</RCPPort>
XXXX is the current RCP port number the VideoJet is listening to

<HTTPPort>XXXX</HTTPPort>
XXXX is the current HTTP port number the VideoJet is listening to

<HTTPSPort>XXXX</HTTPSPort>
XXXX is the current HTTPS port number the VideoJet is listening to

<dhcp>X</dhcp>
X indicates if dhcp is switched on (1) or off (0)
```

<dhcpState>X</dhcpState>
X indicates the current status of dhcp (when switched on). (0:init; 1:IP assigned; 2:no IP obtained, fallen back to default IP; 3: new IP obtained, device going to reboot)

<cameraFrontendID>XXXX</cameraFrontendID>
XXXX is the ID of the (internal) camera frontend (see also Bicom documentation)

<Flags>X</Flags>
X is the flag value returned by RCP command CONF_CM_DEFAULT_CONFIG

<Lines>X</Lines>
X is the number of video lines the device has

<BcJpeg>X</BcJpeg>
X indicates the ability to deliver JPEGs via the broadcast JPEG retrieval mechanism

<unitStatelessIPv6Address>XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX</unitStatelessIPv6Address>
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX is the automatically assigned IPv6 address

<scanID>X</scanID>
X is the ScanID, used for residential cloud services APP to identify the camera in the Wifi network

<privacyMode>X</privacyMode>
X indicates, if privacy mode is switched on (1) or off (0)

<ClusterMember>XXX.XXX.XXX.XXX</ClusterMember>
XXX.XXX.XXX.XXX is the ip address of one of the cluster members; there might be multiple entries including this device

<ClusterMemberMAC>XX-XX-XX-XX-XX-XX</ClusterMemberMAC>
XX-XX-XX-XX-XX-XX is the MAC address of one of the cluster members; there might be multiple entries including this device

<ClusterID>XX</ClusterID>
XX is index number of the above ClusterMember list of this device

<deviceCapabilities>XXXXXXXXXX...XX</deviceCapabilities>
XXXXXXXXXX...XX is a subset of the binary payload of the reply to the command CONF_DEVICE_CAPABILITIES
(each byte is printed with %2x) and can help to identify the availability of features or the capabilities of this device.
The number of bytes given in this section can vary depending on the device and the firmware version.
The following device capability tags are supported: DEVICE_TYPE, PTZ_CAMERA, NBR_VIDEO_IN, PLATFORM_TYP, VIRTUAL_LINES.

Optional Extension: Targeting Autodetect Scans at devices matching certain criteria

The autodetect process can be targeted at a certain group of devices identified by a defined criteria. This sort of autodetect is device-wise only handled, when sent to the configurable UDP port, whose default port number is 1800. It is **not** possible via the static autodetect port 1757. To perform a targeted autodetect scan, an extension needs to be added to the usual autodetect scan command. For these targeted kinds of autodetect scans, it is obligatory that the scan packet consists of an RCP plus protocol header, the autodetect request header and the extension itself. The bytes of this extension are considered in the length field of a preceeding RCP header.

16	32
Scan Extension Identifier 4 Bytes	
Size 4 Bytes	

Scan Extension Identifier

The scan extension identifier determines the criteria that needs to be matched by the scanned devices in order to reply to the request.
Currently the following possible criteria are defined:

MAC address	0x1

Size

The Size field of the extension informs about the number of bytes used in the extension. The number includes the 4 bytes of the Scan Extension Identifier field, the 4 bytes of the Size field itself and the length of the following payload in bytes.

In case of the MAC address extension, the Size field is followed by one or more MAC addresses of 6 bytes length each. A scan extension limiting the autodetect scan to 2 devices, the size field of the scan extension should contain a value of 20. (4 bytes Scan Extension Identifier + 4 bytes Length + 6 bytes first MAC address + 6 bytes second MAC address)

Retrieving thumbnail JPEGs using the Autodetecting Broadcast Mechanism

The Bosch Security Systems products support a mechanism that allows to retrieve thumbnail JPEGs via a broadcast autodetect mechanism. The mechanism requires to send a UDP packet similar to the autodetect scan packet to the configurable autodetect port of the device, whose default port number is 1800. The device replies with a packet containing the thumbnail JPEG. The JPEG retrieval can not be accomplished via the static autodetect port 1757.

Format of the request packet of an autodetect based JPEG request

The request packet triggering the autodetect based JPEG retrieval consists of an RCP plus

protocol header and an autodetect based JPEG request header, followed by a MAC list scan extension.

RCP plus protocol header

The RCP plus protocol header's command id field is set to 0x099e (CONF_JPEG). The data type field is set to 0x0c (P_OCTET). The numeric descriptor field identifies the line of the targeted device, from which the JPEG is to be retrieved. The length field of the request indicates the number of payload bytes and therefore is to be set to the length of the autodetect based JPEG request header plus the length of the scan extension (e.g. 12 bytes for autodetect based JPEG request header, 8 bytes for scan extension header and 6 bytes for MAC address = 26 bytes).

Autodetect based JPEG Request Header

The first four bytes of the autodetect based JPEG request header consist of a static magic number, that is always set to the values given in the table below. The sequence number given in the request is returned in the reply, so that replies can be matched to the respective request.

		16	32
0x4b 1 Byte	**0xa5** 1 Byte	**0xc3** 1 Byte	**0x55** 1 Byte
Sequence Number 4 Bytes			
0x00 1 Byte	**0x00** 1 Byte	**Reply Port** 2 Bytes	
8		24	

Sequence Number

This sequence number must be a randomized number.

Reply Port

Specifies the reply UDP port e.g. RP1 0x06; RP2 0xDE -> reply port: 1758

MAC list scan extension

The autodetect based JPEG request can only be targeted at selected devices. To announce the scope of the request, extending the request packet by a MAC list scan extension, like it is defined in the chapter "Autodetecting Devices" of this document, is obligatory. As the intended use of this JPEG retrieval mechanism is limited to situations where the regular JPEG retrieval mechanisms (HTTP request for the JPEG, RCP request of CONF_JPEG within a regular RCP+ client session, etc.) do not work, the MAC list of the scan extension is limited to 1 MAC address, i.e. only one JPEG from one device can be retrieved at a time.

Format of the reply packet to an autodetect based JPEG request

The reply packet to the request for an autodetect based broadcast JPEG consists of an RCP plus protocol header and an autodetect based JPEG request header, followed by the JPEG data.

RCP plus protocol header

The RCP plus protocol header's command id field is set to 0x099e (CONF_JPEG). The data type field is set to 0x0c (P_OCTET). The numeric descriptor field identifies the line of the targeted device, from which the JPEG was retrieved. The length field of the reply indicates the number of payload bytes and therefore is to be set to the length of the autodetect based JPEG request header plus the length of the JPEG thumbnail.

Autodetect based JPEG Request Header

The autodetect based JPEG request header in the reply is byte for byte a copy of the same header in the respective request.

JPEG thumbnail

The JPEG thumbnail data follows the autodetect based JPEG request header.

Setting an IP Address using Broadcast Mechanism

If the current IP address of a device doesn't match the subnet settings, this unit cannot be reached using the standard Remote Control Protocol. All IP parameters of a VideoJet can be set using a UDP broadcast message to port **1759**.
This is supported for IPv4 broadcast **255.255.255.255**
And also for IPv6 multicast to (link-scope all-hosts multicast) **ff02::1**

Payload Structure (Basic Version IPv4 only)

16 32

Hardware Address 6 Bytes ...	
	Reserved 2 Bytes (shold be zero)
IP Address 4 Bytes	
Subnet Mask 4 Bytes	
Gateway IP 4 Bytes	
Reserved 4 Bytes	

8 24

Hardware Address

MAC address of the device.

IP Address

IP address to set.

Subnet Mask

Subnet mask to set.

Gateway IP

Gateway IP address to set.
As all VideoJets will receive this message, only the unit matching the hardware address (HW1-HW6) will update its IP parameters.

NOTE: There will be no reply to this command.

NOTE: This command will only work if NO service password is set.

Payload Structure (Extended Version IPv4 and IPv6 settings)

16 32

Hardware Address 6 Bytes ...		
	Version 1 Byte	**Ipv6 PrefixLen** 1 Byte
IPv4 Address 4 Bytes		
IPv4 Subnet Mask 4 Bytes		
Gateway IPv4 Address 4 Bytes		
IPv6AddrString 64 Bytes		
IPv6GatewayAddrString 64 Bytes		

8 24

Hardware Address

MAC address of the device.

Version

Version of this command (must be 2)

Ipv6 PrefixLen

IPv6 Prefix length used for the IPv6 Device Address

IPv4 Address

IPv4 Address to set.

IPv4 Subnet Mask

Subnet mask to set.

Gateway IPv4

Gateway IPv4 address to set.

IPv6AddrString

Zero terminated string containig the IPv6 address to set.
Format: e.g. FD00::1234:5678

IPv6GatewayAddrString

Zero terminated string containig the IPv6 gateway address to set.
Format: e.g. FD00::1234:5678
As all VideoJets will receive this message, only the unit matching the hardware address (HW1-HW6) will update its IP parameters.

NOTE: There will be no reply to this command.

NOTE: This command will only work if NO service password is set.

Request or setting of system or network parameters using Broadcast Mechanism

In addition to the SETTING_AN_IP_ADDRESS_USING_BROADCAST, an additional way to setup non network configured devices has been introduced. The device itself is selected using the ethernet mac address as the first parameter inside the command structure. A list of commands including payload length information may be added. All commands must be addressed using UDP broadcast message to port **1760**.

Payload Structure

MAC address 6 Bytes ...	
Reserved 2 Byte	**Command ID** 2 Byte
Payload length 2 Byte	**Variable length payload**

	n Bytes	

MAC address

MAC address of the unit to configure

Command ID

ID of broadcast setting command

LED_BLINK = 0x0000

Payload length 4 bytes. Requests the unit blink the power led; payload containin the number of seconds for blinking. NOTE duration is limited to 60 sec.

SWITCH DHCP = 0x0001

Payload length 4 bytes. Switches DHCP on (payload = 1) or off (payload=0).
NOTE: This command will only work if NO service password is set.

Payload length

Length of the payload section

Variable length payload

Payload; depending on Command ID
NOTE: There will be no reply to this command.

Extension for RCP+ commands containing IPv6 addresses or host names

For transmitting IPv6 addresses or host names in RCP+ Commands there is a string table extension in firmware version 5.50 and later. The layout of the RCP+ commands remains the same as it was in earlier firmware versions and it is compatible to older clients as long as only IPv4 Addresses are transmitted. If IPv6 Addresses are contained in the command then this extension has to be used.

RCP+ Command with string table

RCP Header
RCP Data
String Table Header
String Table Entry 1
String Table Entry 2
...
String Table Entry n

String Table Header

16 32

TableLenBytes 4 Bytes		
Version 1 Byte	**Reserved** 1 Byte	**NumEntries** 2 Bytes

TableLenBytes

Length of hole string table including this header.

Version

Must be 1.

Reserved

Must be 0.

NumEntries

Number of entries in the table.

String Table Entry

16 32

LenBytes 2 Bytes	**Datatype** 1 Byte	**Reseved** 1 Byte
Data Variable lenght		

LenBytes

Length of this entry including this header.

Datatype

Type of content:
0 = IPv4 binary encoded value (always 4 bytes)
1 = IPv6 binary encoded value (always 16 bytes)
2 = HostName or IPv4/IPv6 Address String (encoded as zero terminated string)

Reseved

Must be 0.

Data

n bytes of entry data.

Rules for using the table

- The "StringTable" extension is an option.

- If the RCP-Data contain any IP-Address which current(!) value is other than an IPv4 type then the
StringTable option must be used.

- If the RCP-Data does not contain any IP-Address which current(!) value is other than an IPv4 type then the
StringTable option should NOT be used to be compatible with older firmware versions and with commands which

might not (yet) support this feature.

- If the StringTable is used then:

1. The string_table_avail flag must be set to 1 in the RCP+ Header
2. All IP-values within "RCP-Data" must be stored in the string table (also when they are IPv4 values)
3. The DWORD-field in "RCP-Data" (which normally would store the IPv4 value) has to contain an offset-pointer to the location in the string table where the actual IP-Address-value can be found.
   Offset has to be calculated from "Head of StringTable Header" to "Head of StringTable Entry Header" in Bytes.
4. There is only one StringTable allowed in an RCP+ Packet.
5. The len field of the RCP+ Header must NOT include the size of the StringTable.
6. The len field of the TPKT Header (Framing Header) MUST include the size of the StringTable.
   (This means: RCP-Packet and StringTable are both content of the same TPKT Packet)

RCP Command Notification

A RCP client can register for a special message CONF_RCP_CMD_NOTIFICATION (0xff23) in order to be informed when any other RCP+ command is written to the device.

The message is sent out at a frequency of maxium 100ms and contains the tag codes of all RCP+ commands with the direction 'write' that have been processed by the device since the last message. If no RCP+ command was written within this time, no message will be sent. Besides the tag codes the message contains some more information about the commands as described below.

The payload of the message contains one or multiple elements of the following structure (one element for each RCP command). The maxium number of elements is 32.

Message payload element

16 32

Command Tag 2 Bytes		**Datatype** 1 Byte	**Direction** 1 Byte
Action 1 Byte	**Reserved** 1 Byte	**Client ID** 2 Bytes	
Reserved 4 Bytes			
NumDes 2 Bytes		**Count** 2 Bytes	

8 24

Command Tag

Tag code of the RCP command.

Datatype

Datatype of the RCP command (e.g. P_OCTET).

Direction

Direction of the RCP command (currently only WRITE).

Action

Action of the RCP command (currently only REPLY).

Client ID

Client Id of the RCP client that sent the RCP command.

NumDes

Numeric Descriptor (NumDes) parameter value of the RCP command.

Count

Number of times that this RCP command has been processed.

CONF_RCP_CLIENT_REGISTRATION

Tag code	NumDes	Message	SNMP Support
0xff00	no	no	no

	Datatype	Access Level	Description
Read	%	unprotected	%
Write	p_octet	unprotected	see detailed description

The registration is a standard RCP packet with the command code 0xFF00 with data type P_OCTET (0x0c). The Numeric Descriptor is not used. The Client ID in the header section is ignored.

Payload Structure

Payload Structure

RegType 1 Byte	**Reserved** 1 Byte	**Client ID** 2 Byte	
PE 1 Byte	**PL** 1 Byte	**Nbr** 2 Byte	
Tag 1 2 Byte		N x 2 Bytes	**Tag N** 2 Byte ...

	Char 1 1 Byte	N x 1 Byte	**Char N** 1 Byte

RegType

Values:

Modify Registration	0x00
Normal Registration	0x01
Hook Back Registration	0x03

Client ID

Unused

PE

Password and User Delivery Encryption.

Values:

Plain text	0x00
MD5 hash	0x01

PL

Password and User String Length. Number of bytes to follow for the password string.

Nbr

Number of message tags inside this packet.

Tag 1 - N

Tag codes for the messages which should be passed to the RCP client (see tag code table for capability to generate messages) – A Tag code of 0xFFFF will force the VideoJet to generate a message on all possible events.

Char 1 - N

Password string itself. NOTE: No \0 termination!

Registration Procedure

Normal Registration

1. The client opens up a TCP connection to the VideoJet.
2. The client sends an RCP registration as mentioned above.
3. The RCP server (the VideoJet) will respond with the result code and the assigned Client ID.
4. The client is now allowed to send RCP Requests provided with the Client ID.

Modify Registration

An already registered client can change the message list.

Hook Back Registration

This mechanism is used during connection establishment only. A called host is requested to do a back register to the calling host. To achieve proper client allocation on the calling host, the Client ID of the server session on the called host must be provided in the 'Hook back' registration.

Format: XXXX@PasswordString

XXXX:	The Client ID in hexadecimal notation (4 digits)
PasswordString:	as defined below

Password and User Delivery and Encryption

There are currently three user levels on the videojet:

- live – For video access only; no control
- user – For video access and control over the video stream (e.g. caminput)
- service - for video and all administrative access

Plain text:

The identification string must have the following format: +Username:Password+

MD5 hash encryption:

1. A 16 character random string for MD5 hash calculation must be requested from the VideoJet (RCP command CONF_RCP_REG_MD5_RANDOM).
2. A string +random_string+++username:password+ must be formed.
3. The response MD5 hash over this string must be calculated.
4. The identification string in the register packet must have the following format: +Username:random_string:response_string+

NOTE: No additional whitespaces are allowed.

Server Reply

Result 1 Byte	**Level** 1 Byte	**Client ID** 2 Bytes

8 16 24 32

Result

Values:

Registration failed	0x00
Registration successful	0x01

Level

The level after this registration.

Client ID

Identification number under which the client is now registered at the VideoJet.

CONF_RCP_CLIENT_REGISTRATION_V2

Tag code	NumDes	Message	SNMP Support
0xff04	no	no	no

	Datatype	Access Level	Description
Read	%	unprotected	%
Write	p_octet	unprotected	see detailed description

The registration is a standard RCP packet with the command code 0xFF04 with data type P_OCTET (0x0c). The Numeric Descriptor is not used. The Client ID in the header section is ignored. The difference to CONF_RCP_CLIENT_REGISTRATION is that this command support extra flags.

Payload Structure

Payload Structure

RegType 1 Byte	**Flags** 1 Byte	**Client ID** 2 Byte	
PE 1 Byte	**PL** 1 Byte	**Nbr** 2 Byte	
Tag 1 2 Byte		N x 2 Bytes	**Tag N** 2 Byte ...

	Char 1 1 Byte	N x 1 Byte	**Char N** 1 Byte

RegType

Values:

Modify Registration	0x00
Normal Registration	0x01
Hook Back Registration	0x03

Flags

Values:

MESSAGES_IN_V2_FORMAT. Register for messages with extended timestamp and sequence number header. If this flag is set, then all messages sent from the server to this client have an additional timestamp and sequence number header right after the RCP+ Header. This header is identical to the "Timestamp Header Structure" returned by CONF_RCP_MSG_V2_HISTORY. Please see description of CONF_RCP_MSG_V2_HISTORY for details.	0x02
TAGGED_OPTION_HDR. (Signal that the client will send extra tagged option list on each request. Internal only)	0x04

Client ID

Unused

PE

Password and User Delivery Encryption.

Values:

Plain text	0x00
MD5 hash	0x01

PL

Password and User String Length. Number of bytes to follow for the password string.

Nbr

Number of message tags inside this packet.

Tag 1 - N

Tag codes for the messages which should be passed to the RCP client (see tag code table for capability to generate messages) – A Tag code of 0xFFFF will force the VideoJet to generate a message on all possible events.

Char 1 - N

Password string itself. NOTE: No \0 termination!

Registration Procedure

Normal Registration

1. The client opens up a TCP connection to the VideoJet.
2. The client sends an RCP registration as mentioned above.
3. The RCP server (the VideoJet) will respond with the result code and the assigned Client ID.
4. The client is now allowed to send RCP Requests provided with the Client ID.

Modify Registration

An already registered client can change the message list.

Hook Back Registration

This mechanism is used during connection establishment only. A called host is requested to do a back register to the calling host. To achieve proper client allocation on the calling host, the Client ID of the server session on the called host must be provided in the 'Hook back' registration.

Format: XXXX@PasswordString

XXXX:	The Client ID in hexadecimal notation (4 digits)
PasswordString:	as defined below

Password and User Delivery and Encryption

There are currently three user levels on the videojet:

- live – For video access only; no control
- user – For video access and control over the video stream (e.g. caminput)
- service - for video and all administrative access

Plain text:

The identification string must have the following format: +Username:Password+

MD5 hash encryption:

1. A 16 character random string for MD5 hash calculation must be requested from the VideoJet (RCP command CONF_RCP_REG_MD5_RANDOM).
2. A string +random_string+++username:password+ must be formed.
3. The response MD5 hash over this string must be calculated.
4. The identification string in the register packet must have the following format: +Username:random_string:response_string+

NOTE: No additional whitespaces are allowed.

Server Reply

Result 1 Byte	**Level** 1 Byte	**Client ID** 2 Bytes

16 32

8 24

Result

Values:

Registration failed	0x00
Registration successful	0x01

Level

The level after this registration.

Client ID

Identification number under which the client is now registered at the VideoJet.

CONF_RCP_CLIENT_UNREGISTER

Tag code	NumDes	Message	SNMP Support
0xff01	no	no	no

	Datatype	Access Level	Description
Read	%	unprotected	%
Write	p_octet	unprotected	see detailed description

Request

The unregister is a standard RCP packet with the command code **0xFF01** with no payload section. The needed information (Client ID) will be taken from the header section.

Reply

The reply will have the following format:

Result 1 Byte	**Level** 1 Byte	**Client ID** 2 Bytes

8 16 24 32

Result

Values:

Unregister failed	0x00
Unregister successful	0x01

Level

The level before this unregister.

Client ID

Identification number under which the client was registered at the VideoJet.

NOTE: This command is NOT readable.

CONF_RCP_CLIENT_TIMEOUT_WARNING

Tag code	NumDes	Message	SNMP Support
0xff03	no	yes	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	not supported; will generate message if client idle timeout will happen in 1 minute
Write	f_flag	access_right_minimal	not supported; will generate message if client idle timeout will happen in 1 minute. see detailed description

Timeout handling

The RCP server has to deal with lots of RCP connections at the same time. If an RCP client is going down without notification to the server, an orphan table entry will remain. To avoid this, a timeout mechanism is provided. Each RCP client must show any activity within 10 minutes. After this time the registration of a specific RCP client is invalid. For easy handling, the RCP client can register itself for an RCP_CLIENT_TIMEOUT_WARNIG message. The client will be notified 1 minute before the timeout occurs.

The timeout renewal can be achieved by any RCP read or write command. A good advise is to read the RCP_CLIENT_REGISTRATION.

CONF_RCP_REG_MD5_RANDOM

Tag code	NumDes	Message	SNMP Support
0xff05	no	no	no

	Datatype	Access Level	Description
Read	p_string	unprotected	returns a 16 char. random string to be used in MD5 hash encrypted registration. see detailed description
Write	%	unprotected	%

NOTE: To avoid 'denial of service' attacks, the RCP server will only grant up to 5 random strings at the same time. Maximum time from the request of a random string to its use: 5 seconds. Multiple requests from the same host are ignored.

CONF_RCP_TRANSFER_TRANSPARENT_DATA

Tag code	NumDes	Message	SNMP Support
0xffdd	yes	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	see detailed description
Write	p_octet	access_right_user	see detailed description

The transparent data from and to the serial interfaces is handled by the RCP to achieve reliable transfer of information. To gather control over the remote serial interface a successful registration is necessary.

Request packet

Options 1 Byte	**Reserved** 1 Byte	**Lease Time** 2 Bytes
Trans. Data 1 1 Byte	N x 1 Byte	**Trans. Data N** 1 Byte

Options

Currently no options used

Lease Time

Time in seconds the lease is requested

Values:

Only this packet should be sent out; no further control is requested	0x0000
Indefinite lease time; request lease as long as the current registration is valid	0xFFFF

NOTE: The lease time should be treated as a request; the VideoJet may switch leases before the requested time is over due to a higher prioritized RCP client

For this direction, an RCP Write command to a specific port (addressed by the Numeric Descriptor) is necessary. Once the VideoJet has received a TRANSFER_TRANSPARENT_DATA command, it checks whether the RCP client is in control or not. If the RCP client is allowed to send data to the serial interface, the reply will present an OK. If the RCP client is not allowed to send data, a FAIL will be returned. In this case another RCP client is controlling the serial out. The timeout and priority handling of the serial ports is beyond the scope of this document.

NOTE: This command is NOT readable.

Write reply packet

Code 1 Byte	**Reserved** 1 Byte	**Reserved** 2 Bytes

8 16 24 32

Code

Values:

Access to the serial port denied	0x00
Access to the serial port granted	0x01

Reading the TRANSFER_TRANSPARENT_DATA command

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 the serial port.

NOTE: Despite a positive reply to a read command, the port may be locked by another RCP client in the time slice between the read and a following write command.

Serial IN -> RCP Client

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. Data is posted if the corresponding RCP client is in control only.

If no client has a lease on the serial port, a message to all registered clients will be generated.

NOTE: The received message will carry NO header.

CONF_CONNECT_PRIMITIVE

Tag code	NumDes	Message	SNMP Support
0xff0c	no	no	no

	Datatype	Access Level	Description
Read	%	access_right_live	%
Write	p_octet	access_right_live	see detailed description

Switching Media Channels

Each RCP client may have different media sources and media receivers. To connect these media channels, a low level connect/disconnect primitive is implemented. These low level primitives can be grouped together to more complex commands, which can connect more than one media channel at the same time.

The primitives are designed to be asserted by media sources or receivers itself. Manager software should use the high level commands (see next chapter).

Using GET

By sending a CONNECT GET to the remote endpoint, this endpoint will start sending media to the requested address. For example, a VideoJet receiver is registered at a VideoJet transmitter. On sending a 'CONNECT GET video to port 1234' to the transmitter, the transmitter starts to send video to the appropriate port at the receiver.

The endpoint requesting the media stream (the VideoJet receiver) must be prepared to receive the stream from the network.

The coding bits for video must be set according to the receivers capability to decode the data stream. E.g. a Decoder capable of decoding MPEG2 and MPEG4 should set all these bits.

Using PUT

By sending a CONNECT PUT to the remote endpoint, the endpoint will prepare itself for receiving data from the one initiating the request. For example, a VideoJet transmitter is registered at a VideoJet receiver. On sending a 'CONNECT PUT video to port 2222' to the receiver, the receiver will prepare its network stack to receive data from the appropriate port.

The endpoint initiating the media stream (the VideoJet transmitter) must start sending the media stream to the network.

The coding bits for video must be set to a value suitable to the current encoding parameters inside the VideoJet. E.g. if a video encoder is set to encode MPEG4, only a PUT request having these bits set is accepted.

The coder parameter is used to address a specific encoder/decoder inside the VideoJet. As the VideoJet may have a Dual-stream feature, one physical line input might be connected to more than one encoder. By setting a coder to another value than 0, the VideoJet only tries this specific coder to establish a connection. When the coder value is set to 0, all coders inside the system will be checked.

Each connect primitive will result in a Session ID for maintaining and controlling the media stream with other RCP commands. The Session ID can be treated as unique on the entire system.

DISCONNECT

By sending a DISCONNECT supplied with a valid Session ID, the receiver of the command will stop sending/receiving to/from the media stream. The endpoint initiating the DISCONNECT command must stop receiving/sending from/to the media stream.

Request Packet

Sequence of:

Method 1 Byte	**Media** 1 Byte	**Reserved** 1 Byte	**Flags** 1 Byte
Reserved 4 Bytes			
Media Descriptor 1 M Bytes (Length depends on Media type)			

N x M Bytes	**Media Descriptor N** M Bytes ...

Method

Values:

GET	0x00
PUT	0x01
Key transport	0xE0

Media

Values:

Video	0x01
Audio	0x02

Data	0x03

Flags

Bit1: Do reverse Login. This will control whether the called server will perform a reverse RCP Login.

Bit2: Free Running Connection. When this bit is set, the connection does not need to be re-triggered by the command CONF_RCP_CONNECTIONS_ALIVE.

Bit3: Optional signalling for streaming (for correct connection list display)

Bit4: Media key request (when using GET) / Media key is appended at the end of this section (when using PUT)

Bit5: Dry Connect. If this bit is set a replay connection is established (and a session id to control the replay connection is returned). The replay source is not connected. This can be done using the CONF_RCP_CONNECT_SALVO command

Bit6: Transcoder: Set this bit to instanciate and establish a connection to the transcoder

Bit7: Media over ssl: This Bit is set in the response to signal that the device supports "media over ssl".

Bit8: Multi ROI capable: This Bit signals that a client is able to provide a session ID to steer multiple ROIs (region of interrest) with E-PTZ.

Media Descriptors 1 - N

Video

MEP 1 Byte	**Flags** 1 Byte	**MTA Port** 2 Bytes
MTA IP Address 4 Bytes		
Coder 1 Byte	**Line** 1 Byte	**MCTA Port** 2 Bytes
MCTA IP Address 4 Bytes		
Coding 2 Bytes		**Resolution** 2 Bytes
Linked Coder 1 Bytes	**Linked Line** 1 Bytes	**Flags2** 2 Bytes

16 32

8 24

MEP

Media Encapsulation Protocol.

Values:

RTP over UDP	0x01
RTP over TCP	0x04

Note: When using RTP over TCP, media channels must be established as described in 'RCP Protocol Procedure / Receiving media data'.

Flags

Values:

Bit1	Substitute connection (if an RX connection to the requested interface is already established, the old connection will be disconnected and substituted by a new connection coming with this request).
Bit2	Use relative to line coder addressing.
Bit3	Flags2 are used
Bit7	Request an exclusive (ROI) stream. (Only applicaple if device is in multi ROI mode and an exclusive coder is still available).

MTA Port

Media Transport Address - Network port.

MTA IP Address

Media Transport Address IP Address

Values:

VideoJet will take address from the incoming TCP header	0.0.0.0
VideoJet will take its configured multicast group IP	224.0.0.0

Coder

Specifies the internal coder number; when set to 0, the appropriate coder corresponding to the given line number and/or coding parameter is chosen by the VideoJet itself.

Line

Identifies the Line Input/Output channel for the specified coding engine.

MCTA Port

Media Control Transport Address Network Port. Currently unused.

MCTA IP Address

Media Control Transport Address IP Address. Currently unused.

Coding

Values:

Bit3	MPEG-4 (elementary stream)
Bit4	MPEG-2 (only video)
Bit5	Meta Data (output from motion detector…)
Bit7	H264
Bit8	JPEG
Bit10	H265
Bit15	Recorded Media
Bit16	MPEG-2 (program stream)

Resolution - (obsolete, not checked any more)

Values:

Bit1	QCIF

Bit2	CIF
Bit3	2CIF
Bit4	4CIF
Bit6	QVGA
Bit7	VGA
Bit8	HD720
Bit9	HD1080
Bit10	WD144
Bit11	WD288
Bit12	WD432

Linked Coder

Use this field for signalling the callers local coder in the connection list of the called device.

Linked Line

Use this field for signalling the callers local line in the connection list of the called device.

Flags2

Only evaluated if Bit3 of Flags is set.

16 32

MEP 1 Byte	**Flags** 1 Byte	**MTA Port** 2 Bytes
MTA IP Address 4 Bytes		
Coder 1 Byte	**Line** 1 Byte	**MCTA Port** 2 Bytes
MCTA IP Address 4 Bytes		
Coding 2 Bytes		**Reserved** 2 Bytes
Linked Coder 1 Bytes	**Linked Line** 1 Bytes	**Reserved** 2 Bytes

8 24

MEP

Media Encapsulation Protocol.

Values:

RTP over UDP	0x01
RTP over TCP	0x04

Flags

Values:

Bit1	Substitute connection (if an RX connection to the requested interface is already established, the old connection will be disconnected and substituted by a new connection coming with this request).
Bit2	Use relative to line coder addressing.

MTA Port

Media Transport Address - Network port.

MTA IP Address

Media Transport Address IP Address

Values:

VideoJet will take address from the incoming TCP header	0.0.0.0
VideoJet will take its configured multicast group IP	224.0.0.0

Coder

Specifies the internal coder number; when set to 0, the appropriate coder corresponding to the given line number and/or coding parameter is chosen by the VideoJet itself.

Line

Identifies the Line Input/Output channel for the specified coding engine.

MCTA Port

Media Control Transport Address Network Port. Currently unused.

MCTA IP Address

Media Control Transport Address IP Address. Currently unused.

Coding

Values:

Bit1	G.711 (sampling rate: 8 kHz, rtp clock rate: 90 kHz)
Bit2	AAC
Bit3	G.711_8kHz (sampling rate: 8 kHz, rtp clock rate: 8 kHz)
Bit4	L16_16kHz (sampling rate: 16 kHz, rtp clock rate: 16 kHz)
Bit5	L16 (sampling rate: 16 kHz, rtp clock rate: 90 kHz)
Bit15	Recorded Media
Bit16	MPEG-2 (program stream)

Linked Coder

Use this field for signalling the callers local coder in the connection list of the called device.

Linked Line

Use this field for signalling the callers local line in the connection list of the called device.

Note: Audio packetzing

For audio G.711, the encapsulation of bitstream RTP packets are required to be 640 bytes (equvalent of 80ms samples).

Data

16 32

MEP 1 Byte	**Reserved** 1 Byte	**Reserved** 2 Bytes
MTA IP Address 4 Bytes		
Coder 1 Byte	**Line** 1 Byte	**Reserved** 2 Bytes
Reserved 4 Bytes		

Coding 2 Bytes		**Reserved** 2 Bytes
Linked Coder 1 Bytes	**Linked Line** 1 Bytes	**Reserved** 2 Bytes

8 24

MEP

Media Encapsulation Protocol.

Values:

RCP Intrinsic	0x02

MTA IP Address

Media Transport Address IP Address

Values:

VideoJet will take address from the incoming TCP header	0.0.0.0

Coder

Informational.

Line

Identifies the Line Input/Output channel for the specified coding engine.

Coding

Values:

Bit1	RCP
Bit15	Recorded Media

Linked Coder

Use this field for signalling the callers local coder in the connection list of the called device.

Linked Line

Use this field for signalling the callers local line in the connection list of the called device.

Span Replay

16 32

MEP 1 Byte	**Reserved** 1 Byte	**Media Handle** 2 Byte
Target ID 4 Bytes		
Coder 1 Byte	**Line** 1 Byte	**Reserved** 2 Byte
Target Idx 1 Byte	**Lun Idx** 1 Byte	**Span Idx** 2 Byte
Coding 2 Byte		**Resolution** 2 Byte
Reserved 2 Byte		**Bandwidth** 2 Byte

8 24

MEP

Media Encapsulation Protocol.

Values:

API Callback	0x06

Media Handle

Identifier that is associated with the rtp data and is supplied with the rtp data in the callback function.

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

Coder

Specifies the internal coder number; when set to 0, the appropriate coder corresponding to the given line number and/or coding parameter is chosen by the VideoJet itself.

Line

Identifies the Line Input/Output channel for the specified coding engine.

Target Idx

The index of the target on the iSCSI Server. This index can be obtained by sending a disconver to the server.

Lun Idx

The index of the Lun of the specified target on the iscsi server.

Span Idx

The index of the span.

Coding

Values:

Bit15	Recorded Media

Resolution - (obsolete, not checked any more)

Values:

Bit1	QCIF
Bit2	CIF
Bit3	2CIF
Bit4	4CIF
Bit6	QVGA
Bit7	VGA
Bit8	HD720
Bit9	HD1080
Bit10	WD144
Bit11	WD288
Bit12	WD432

Bandwidth

Initial bandwidth request in kBit/s Must have an value > 0. Only used on VJX00 series.

NOTE: If this command is used with the vj api, an specific error of the following type may be returned

SPAN_ERR_INTERNAL	0x01
SPAN_ERR_INV_SPN_IDX	0x02
SPAN_ERR_INV_HDR_TYPE	0x03
SPAN_ERR_NOT_MNTD	0x05
SPAN_ERR_INV_FS	0x06
SPAN_ERR_INV_LUN_NFO	0x07
SPAN_ERR_BAD_HDR_CKSM	0x08
SPAN_ERR_RD_ONLY	0x0a
SPAN_ERR_NO_REC_DAT	0x0b
SPAN_ERROR_INV_PART_NFO	0x0c
ISCSI_ERROR_CONNECT	0x31
ISCSI_ERROR_INV_LUN	0x33
ISCSI_ERROR_LOGIN	0x34
ISCSI_ERROR_INV_TRG_IDX	0x35

Key transport

When the media key bit is selected in the header flags, a variable length key field is appended at the end of the media descriptor. The key field has the following structure

Sequence of:

method 1 Byte	**bytes valid** 1 Byte	**status tag** 1 Byte	**reserved** 1 Byte
key data 4 Bytes			

16 32 8 24

method

must be set to 'Key transport' 0xE0

bytes valid

the number of valid key bytes in this header

status tag

must be set to 'Key transport tag' 0xE0

key data

the key payload

The media key is transferred using a sequence of key headers all with the method field set to 0xE0. The end of the key section is marked by a header with less than 4 bytes (or zero) indicated in the bytes valid field.

The typical key length is 16 bytes (128 bit AES). In this case, four full key headers followed by one empty header must be transferred (4 x 4 bytes valid + 1 x 0 bytes valid = 16 bytes)

For backward compatibility, the header method 0xE0 will be skipped by units with former firmwares versions.

The key is located and stored on each media source (using the CONF_CRYPT_KEY* commands); When the system is configured to publish its keys (see CONF_PUBLISH_MEDIA_KEYS), then the media sources must fill in the secret by sending (PUT method) or replying to (GET method) the connect primitive.

For example, when sending a GET connect primitive to a device and the key transport is selected, then the device will fill in its key in the reply to the connect primitive.

NOTE: be sure to use a secure RCP connection, the keys in the connect primitive are tranferred in plain

Reply Packet

Sequence of:

Method 1 Byte	**Media** 1 Byte	**Status** 1 Byte	**Flags** 1 Byte
Reserved 4 Bytes			
Media Descriptor 1 M Bytes (Length depends on Media type)			

N x M Bytes	**Media Descriptor N** M Bytes ...

Method

Values:

GET	0x00
PUT	0x01

Media

Values:

Video	0x01
Audio	0x02
data	0x03

Status

Values:

Interface not available (Remark: former version may return this error in case of unknown method/media type)	0x00
Access to this interface granted	0x01
Access to this interface rejected	0x02
Session ID invalid or connection no longer active	0x03
Coding parameters incompatible	0x04
Interface data will be supplied by another media descriptor	0x05
The method field in the header does not carry a known method	0x08
The media field in the header does not carry a known media type	0x09

NOTE: Any wildcards used in the transport addresses will be replaced by the real used network addresses.

This command is NOT readable; to gain information on active connection refer to the ACTIVE_CONNECTION_LIST command.

NOTE: The Media Control sockets have currently no function

CONF_DISCONNECT_PRIMITIVE

Tag code	NumDes	Message	SNMP Support
0xff0d	no	yes	yes

	Datatype	Access Level	Description
Read	%	access_right_live	%
Write	p_octet	access_right_live	see detailed description

Payload Structure

16	32
Status 1 Byte	**Cause** 1 Byte	**Reserved** 2 Bytes
Remote Host IP 4 Bytes	

8 24

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.

CONF_CONNECT_TO

Tag code	NumDes	Message	SNMP Support
0xffcc	no	yes	yes

	Datatype	Access Level	Description
Read	%	access_right_live	%
Write	p_octet	access_right_live	see detailed description

Write Packet

16 32

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

8 24

Destination IP Address

The reception of this command will force the host to connect to the mentioned destination IP address.

Line

OBSOLETE
This parameter carries the number of the remote input (for get channels) or the local input (for put channels); For connection to the HDD, the Line byte must carry the partition number of the HD.
This parameter is updated by the local line/coder and remote line/coder settings.

Flags

NOTE: Only one video standard (MPEG2/4) can be used; setting all bits will result in best currently available mode.

Values:

Bit0	Substitute an existing connection
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 a recorded media stream
Bit6	Connect a VCA meta data stream
Bit7	Supress automatic media key exchange
Bit8	Use SSL for the RCP control connection; if no destination port is specified, the remote port defaults to 443 (HTTPS) in case of SSL is requested otherwise 80(HTTP)
Bit9	Request video mode H.264
Bit10	Request video mode JPEG
Bit11	Allow audio mode L16 (setting this bit to 0 uses G711 (default, previous behaviour), setting to 1 allows G.711 and L16 - both with 90kHz timestamp clock)
Bit12	Allow audio mode AAC
Bit13	Request video mode H.265

Local Coder

This parameter carries the number of the local coder number. A wildcard of '0' will result in first match.

Local Line

This parameter carries the number of the local video line number. A wildcard of '0' will result in first match.

Put Channels

Values:

Bit0	Video
Bit1	Audio
Bit2	Data
Bit3-Bit15	Reserved

Remote Coder

This parameter carries the number of the remote coder number (at the destination IP). A wildcard of '0' will result in first match.

Remote Line

This parameter carries the number of the remote video line number (at the destination IP). A wildcard of '0' will result in first match.

Get Channels

Values:

Bit0	Video
Bit1	Audio
Bit2	Data
Bit3-Bit15	Reserved

NOTE: For audio connections, the local and remote line parameter are taken from the video settings.

If this command is extended with the optional appendix, the remote port number for RCP login must be specified. This can be either 1756 for the normal RCP port or any available HTTP port at the remote host. When a port number different to 1756 is used, the login will use a HTTP tunnelling.

Optional Appendix

16		32
Dest. Port 2 Bytes	**Flags** 2 Bytes	
Reserved 2 Bytes	**remote password level** 1 Byte	**remote password length** 1 Byte
remote password 128 Bytes		
8	24	

Dest. Port

This parameter carries the number of the remote TCP port number used for RCP login.

Flags

Values:

Bit0	remote password field is url (supports urls like rtsp://user:pw@160.10.9.222:5540/rtsp_tunnel)

remote password level

remote user: password level: 1-user, 2-service, 3-live, 0-no protection (optional parameter)

remote password length

remote user: password length (optional parameter)

remote password

remote user: password (optional parameter)

Reply / Read / Message

The reply to this command will present the same packet format as the outgoing. The result will contain the really established channels.

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.

CONF_ACTIVE_CONNECTION_LIST

Tag code	NumDes	Message	SNMP Support
0xffc1	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	see detailed description
Write	p_octet	access_right_user	see detailed description

Payload Structure

Sequence of:

16 32

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

8 24

Destination IP Address

IP address to which the unit is connected.

Local Coder

The local connected coder (relative to line).

Local Line

The remote connected line

Flags

Values:

Bit0	Connection is MPEG2 VES
Bit1	Connection is MPEG2 PRG
Bit2	Connection is MPEG4
Bit3	Connection is to HDD replay
Bit4	Connection is started as streaming
Bit5	Connection is using multicast
Bit6	Connection is using TCP for transportation
Bit7	Data stream is encrypted
Bit8	Connection is H.264
Bit9	Connection is Jpeg
Bit10	Connection is using transcoder
Bit11	Connection is H.265

Session ID

Session identifier.

Remote Coder

The remote connected coder (relative to line), (absolute) transcoder if connection is using transcoder

Remote Line

The remote connected line.

TX Channels

See CONNECT_TO command for bit mask.

RX Channels

See CONNECT_TO command for bit mask.

NOTICE: If a session ID is provided in the rcp header, then only the connection list of this specific session ID is supplied.

CONF_MEDIA_SOCKETS_COMPLETE

Tag code	NumDes	Message	SNMP Support
0xffc7	-	yes (Message only)	no

	Datatype	Access Level	Description
Read	%	access_right_user	%
Write	-	access_right_user	sends out a message when all TCP tunneled sockets of a session are connected

CONF_RCP_CONNECTIONS_ALIVE

Tag code	NumDes	Message	SNMP Support
0xffc2	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_live	retriggers a running connection identified by its session_id; returns the current number of active connections
Write	%	access_right_live	%

CONF_CAPABILITY_LIST

Tag code	NumDes	Message	SNMP Support
0xff10	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	%	access_right_minimal	%

Reply payload Structure

0xBABA 2 Bytes	**Version** 2 Bytes	**NbSection** 2 Bytes	**Section 1**	...	**Section N**

Version

current version of the capabilities (0x0001)

NbSection

Number of following sections

Section Structure

Type 2 Bytes	**Size** 2 Bytes	**NbElement** 2 Bytes	**Element 1**	...	**Element N**

Type

Type of Element

Values:

Video	0x0001
Audio	0x0002
Serial	0x0003
IO	0x0004
Camera Data	0x0005

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

For section type Video

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

Type

is one of the following

Values:

VIDEO_ENCODER	0x0001
VIDEO_DECODER	0x0002
VIDEO_TRANSCODER	0x0003

Identifier

Identifier is the RCP numeric descriptor to use to address the entity. It should be unique when associated with the type and the compression. In case of transcoders this numeric descriptor is informative only. It is not necessary to address the entity directly (use the relevant rcp commands instead).

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

Notice: please use CONF_CODER_VIDEO_OPERATION_MODE to configure the video standard. The dependencys are explained in detail at CONF_CODER_VIDEO_OPERATION_MODE.

InputNo

InputNo is the number of the physical input from which the entity gets or puts its video. In case of transcoder this value is unused and set to 0;

Resolution

is one or multiple of the following

Values:

VIDEO_RESO_QCIF	0x0001
VIDEO_RESO_CIF	0x0002
VIDEO_RESO_2CIF	0x0004
VIDEO_RESO_4CIF	0x0008
VIDEO_RESO_CUSTOM	0x0010
VIDEO_RESO_QVGA	0x0020
VIDEO_RESO_VGA	0x0040
VIDEO_RESO_HD720	0x0080
VIDEO_RESO_HD1080	0x0100
VIDEO_RESO_WD144	0x0200
VIDEO_RESO_WD288	0x0400
VIDEO_RESO_WD432	0x0800
VIDEO_RESO_HD2592x1944	0x1000

Notice: this list is not further maintained/extended. If additional resolutions are supported, the VIDEO_RESO_CUSTOM value is set and the available resolutions can be obtained via the command CONF_VID_H264_ENC_BASE_OPERATION_MODE_CAPS.
For JPEG encoders the available resolutions can be obtained via the command CONF_JPEG_STREAM_SETUP_OPTIONS_VERBOSE.

Element Structure

For section type Audio

Type	Identifier	Compr
2 Bytes	2 Bytes	2 Bytes

Type

is one of the following

Values:

AUDIO_ENCODER	0x0001
AUDIO_DECODER	0x0002

Identifier

Identifier is the RCP numeric descriptor to use to address the entity. It should be unique when associated with the type and the compression.

Compression

is one or multiple of the following

Values:

AUDIO_COMP_MPEG2	0x0001
AUDIO_COMP_G711	0x0002
AUDIO_COMP_AAC	0x0004

Element Structure

For section type Serial

Type	Identifier
2 Bytes	2 Bytes

Type

is one of the following

Values:

SERIAL_RS232	0x0001
SERIAL_RS485	0x0002
SERIAL_RS422	0x0004

Identifier

is the RCP numeric descriptor to use to address the entity.

Element Structure

For section type IO

Type	Identifier
2 Bytes	2 Bytes

Type

is one of the following

Values:

IO_INPUT	0x0001
IO_OUTPUT	0x0002
IO_VIRTUAL_IN	0x0003

Identifier

is the RCP numeric descriptor to use to address the entity.

Element Structure

For section type Camera Data

Type	InputNo
2 Bytes	2 Bytes

Type

is one of the following

Values:

CAMDATA_BILINX	0x0001

InputNo

InputNo is the number of the physical video input.

CONF_RCP_CODER_LIST

Tag code	NumDes	Message	SNMP Support
0xff11	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	%	access_right_minimal	%

This command is used to retrieve the list of encoders and decoders from a VideoJet. There are two reply formats supported, the compact and the extended list view. The compact view only reports the absolute coder numbers available on a given line input or output. The extended view offers more information about coding capabilities and other parameters of a coder.

Request payload Structure

16

MediaType	Direction	Flags
1 Byte	1 Byte	1 Byte

8 24

MediaType

Values:

Video	0x01
Audio	0x02
Data	0x03

Direction

physical port is

Values:

Input	0x00
Output	0x01

Flags

one or multiple of

Values:

Request compact format	0x00
Reqeust extended format list	0x01

Compact format reply payload Structure

CoderNbr 1	N x 1 Byte	CoderNbr N
1 Byte		1 Byte

CoderNbr 1 - N

List of the absolute coder numbers

Extended format video reply payload structure

Sequence of

16 32

CodingNbr 2 Byte	**Coding Capabilities** 2 Byte
Coding Current 2 Byte	**Resolution Capabilities** 2 Byte
Resolution Current 2 Byte	**Reserved** 2 Byte
Reserved 4 Byte	

8 24

CoderNbr

The absolute coder number

Coding Capabilities

All coding capabilities is one or multiple of

Values:

H.263	0x0002
Mpeg4	0x0004
Mpeg2	0x0008
Meta Data	0x0010
H.263-1998	0x0020
H.264	0x0040
Jpeg	0x0080
H.265	0x0200
Recorded Media	0x4000
Mpeg2 PrgStr	0x8000

Coding Current

The current available coding capabilities (definitions see above)

Resolution Capabilities

All resolution capabilities is one or multiple of

Values:

QCIF	0x0001
CIF	0x0002
2CIF	0x0004
4CIF	0x0008
Custom	0x0010
QVGA	0x0020
VGA	0x0040
HD720	0x0080
HD1080	0x0100
WD144	0x0200
WD288	0x0400
WD432	0x0800
HD5M	0x1000

Resolution Current

The current available resolution capabilities (definitions see above)

Extended format audio reply payload structure

Sequence of

16	32
CodingNbr 2 Byte	**Coding Capabilities** 2 Byte
Coding Current 2 Byte	**CodParameter Capabilities** 2 Byte
CodParameter Current 2 Byte	**Reserved** 2 Byte
Reserved 4 Byte	
8	24

CoderNbr

The absolute coder number

Coding Capabilities

All coding capabilities is one or multiple of

Values:

G.711	0x0001
AAC	0x0002
G.711 (8kHz timestamps)	0x0004
L16 (16kHz timestamps)	0x0008
L16	0x0010
Recorded Media	0x4000
Mpeg2 PrgStr	0x8000

Coding Current

The current available coding capabilities (definitions see above)

Extended format data reply payload structure

Sequence of

16	32
CodingNbr 2 Byte	**Coding Capabilities** 2 Byte
Coding Current 2 Byte	**CodParameter Capabilities** 2 Byte
CodParameter Current 2 Byte	**Reserved** 2 Byte
Reserved 4 Byte	
8	24

CoderNbr

The absolute coder number

Coding Capabilities

All coding capabilities is one or multiple of

Values:

RCP	0x0001

Coding Current

The current available coding capabilities (definitions see above)

CodParameter Capabilities

All coding CodParameter capabilities is one or multiple of

Values:

RS232	0x0001
RS485	0x0002
RS422	0x0004

CodParameter Current

The current available coding capabilities (definitions see above)

CONF_PUBLISH_MEDIA_KEYS

Tag code	NumDes	Message	SNMP Support
0x09fc	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	switches wheather the media key of a media source are readable and transferred during the hw-hw connection setup
Write	f_flag	access_right_service	switches wheather the media key of a media source are readable and transferred during the hw-hw connection setup

CONF_CRYPT_KEYAUDIO_ENC

	Tag code	NumDes	Message	SNMP Support
	0x09fe	coder instance	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	get key for AES Audio data encryption/decryption
Write	p_octet	access_right_service	set key for AES encryption/decryption of audio data see detailed description

Payload Structure

Key

The 16 Byte Masterkey for AES encryption/decryption of data

Comments:

- This description is valid for the the commands:
  CONF_CRYPT_KEYVIDEO_ENC,
  CONF_CRYPT_KEYVIDEO_DEC,
  CONF_CRYPT_KEYAUDIO_ENC,
  CONF_CRYPT_KEYAUDIO_DEC
- The numeric descriptor contains the number of the coder instance (absolute)
- Set all Key- and all Salt- Bytes to zero, if no encryption/decryption is wished.

CONF_CRYPT_KEYVIDEO_ENC

	Tag code	NumDes	Message	SNMP Support
	0x09ed	coder instance	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	get key for AES Video dataencryption/decryption
Write	p_octet	access_right_service	set key for AES encryption/decryption of video data see detailed description

Payload Structure

16	32
Key 16 Bytes ...	
8	24

Key

The 16 Byte Masterkey for AES encryption/decryption of data

Comments:

- This description is valid for the the commands:
  CONF_CRYPT_KEYVIDEO_ENC,
  CONF_CRYPT_KEYVIDEO_DEC,
  CONF_CRYPT_KEYAUDIO_ENC,
  CONF_CRYPT_KEYAUDIO_DEC
- The numeric descriptor contains the number of the coder instance (absolute)
- Set all Key- and all Salt- Bytes to zero, if no encryption/decryption is wished.

CONF_CRYPT_KEYAUDIO_DEC

	Tag code	NumDes	Message	SNMP Support
	0x09ff	coder instance	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	get key for AES Audio data encryption/decryption
Write	p_octet	access_right_service	set key for AES encryption/decryption of audio data see detailed description

Payload Structure

Key

The 16 Byte Masterkey for AES encryption/decryption of data

Comments:

- This description is valid for the the commands:
  CONF_CRYPT_KEYVIDEO_ENC,
  CONF_CRYPT_KEYVIDEO_DEC,
  CONF_CRYPT_KEYAUDIO_ENC,
  CONF_CRYPT_KEYAUDIO_DEC
- The numeric descriptor contains the number of the coder instance (absolute)
- Set all Key- and all Salt- Bytes to zero, if no encryption/decryption is wished.

CONF_CRYPT_KEYVIDEO_DEC

	Tag code	NumDes	Message	SNMP Support
	0x0a00	coder instance	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	get key for AES Video dataencryption/decryption
Write	p_octet	access_right_service	set key for AES encryption/decryption of video data see detailed description

Payload Structure

16	32
Key 16 Bytes ...	
8	24

Key

The 16 Byte Masterkey for AES encryption/decryption of data

Comments:

- This description is valid for the the commands:
  CONF_CRYPT_KEYVIDEO_ENC,
  CONF_CRYPT_KEYVIDEO_DEC,
  CONF_CRYPT_KEYAUDIO_ENC,
  CONF_CRYPT_KEYAUDIO_DEC
- The numeric descriptor contains the number of the coder instance (absolute)
- Set all Key- and all Salt- Bytes to zero, if no encryption/decryption is wished.

CONF_CRYPT_KEY_GENERATE_ALL

Tag code	NumDes	Message	SNMP Support
0x0a13	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	returns flag=0 when all key are cleared otherwise flag=1
Write	f_flag	access_right_service	generates a random key(flag=1) or clears all(flag=0) keys for AES encryption for all media encoders

CONF_RCP_MSG_V2_HISTORY

Tag code	NumDes	Message	SNMP Support
0x0c57	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	read the list of RCPv2 Event-Messages pending in device history buffer, see detailed description
Write	p_octet	access_right_service	write not supported

Payload structure of request mode 0

16	32
Mode 2 Byte	**MaxResultMsg** 2 Byte
MsgSeqNo 4 Bytes	
BootCount 2 Byte	
8	24

Mode

Select the mode to read messages indicated by a initial sequence number.

Values:

Read by sequence number 0x0

MaxResultMsg

Maximum number of result messages to be returned by one execution of this command

Values:

MsgSeqNo

The first returned message will be the next in device buffer after those indicated by this number. e.g. if MsgSeqNo is 7, then first result is 8 or even a higher number if there is no message 8 according to filters in this data path. If value 0 is supplied then the result starts with the oldest message which is still available on the device. To perform a chunked read the last MsgSeqNo returned by prior request to this command can be the argument of the next request to retreive the next chunk.

Values:

BootCount

Boot counter. This in combination with MsgSeqNo makes a unique ID for a message. If supplied BootCount is lower than actual BootCount then all messages in Buffer are returned up to MaxResultMsg

Values:

Payload structure of request mode 1

16	32
Mode 2 Byte	**MaxResultMsg** 2 Byte
LastNumSeconds 4 Bytes	
8	24

Mode

Select the mode to read messages indicated by by relative timestamp

Values:

Read by relative time 0x1

MaxResultMsg

Maximum number of result messages to be returned by one execution of this command

Values:

LastNumSeconds

The first returned message will be those indicated by this number of seconds back in the past. E.g. value of 30 will start returning the oldest messages which was triggered 30 seconds ago, and then continue with all subsequent up to MaxResultMsg.

Values:

Result Payload

Result Payload Structure

The result payload consists of a repeated sequence for each returned message.
A RCP-Plus header, a Timestamp Header, and then the usual RCP-Message payload.

Timestamp Header Structure

<table>
<tr><th colspan="3">16 32</th></tr>
<tr><td colspan="3">MsgSeqNo
4 Bytes</td></tr>
<tr><td colspan="3">SecSince2000
4 Bytes</td></tr>
<tr><td>Milliseconds
2 Byte</td><td colspan="2">BootCount
2 Byte</td></tr>
<tr><td>SkippedSeqNo
2 Byte</td><td>Timezone
1 Byte</td><td>Reserved
1 Byte</td></tr>
<tr><td colspan="3">8 24</td></tr>
</table>

MsgSeqNo

Message counter starts with 0 on each bootup

Values:

SecSince2000

Time in seconds from RTC in UTC

Values:

Milliseconds

Millisecond extension to SecSince2000

Values:

BootCount

Number of reboots. Use in combination with MsgSeqNo to get a unique Message-ID

Values:

SkippedSeqNo

Number of skipped sequence numbers in this stream due to registering/filtering rules

Values:

Timezone

Timezone in 15 minute steps. Negative values have the leading bit 7 set.
e.g. 0x03 = +45 minutes; 0x84 = -60 minutes

Values:

Reserved

Should be ignored

Values:

CONF_RCP_CONNECT_SALVO

Tag code	NumDes	Message	SNMP Support
0xff13	yes (0 - nothing, 1 to 4 refering to entries of CONF_ADD_REMOTE_DEVICE)	no	no

	Datatype	Access Level	Description
Read	%	access_right_live	%
Write	p_octet	access_right_live	see detailed description

Payload Structure

flags 8 Bits	line 8 Bits	dec inst 8 Bits	relative coder 8 Bits
port 16 Bits		ssl port 16 Bits	
vrm track id 16 Bits		reserved 2 Bytes	
url len 16 Bits		url ...	

url (url len) bytes ...

pwd len 16 Bits	pwd ...

pwd (pwd len) bytes...

flags

bit 0: live video, bit 1: use TCP to get live video, bit 2: internal used, bit 3: replay live only, bit 4: internal used, bit 5: internal used, bit 6: delete cam connected to dec inst (multi view only),

line

line from 1 to max lines (line 0 is mapped to line 1)

dec inst

dec inst from 1 to max ... (dec inst 0 means any) for multiview capable de/transcoders

relative coder

relative coder number from 1 to ... (coder number 0 is mapped to coder number 1)

port

port

ssl port

ssl port

vrm track id

track ID (VRM case)

url len

length of the following url (actual limit is 64)

url

url for the connection salvo, ascii string including zero termination, if the special url starting with "trackid://" is used, the following decimal value is the VRM Track ID (e.g. "tackid://15096")

pwd len

length of the following password (actual limit is 64)

pwd

password for the salvo connection, ascii string including zero termination

CONF_RCP_CONNECT_SALVO

A replay connection can be established in two steps: 1) build up a "dry" (replay) connection by setting the flag in the CONNECT_PRIMITIVE request. A Rcp connection is established and a session id to control the connection is returned. Data is not sent out as long as no source is connected. 2) connect a source to the connection (only implemented for replay) using this RCP_CONNECT_SALVO_COMMAND with the provided session id. The connection will retrieve the data from the specified device If a source is already connected a second call of this command will replace the first connection. The num parameter can be used to refere to an entry from CONF_ADD_REMOTE_DEVICE (num 1 to 4). In that case only a shorter payload (line and relative coder) is required. All other parameters will be taken from the CONF_ADD_REMOTE_DEVICE command. For port and ssl port the first two ports of a CONF_ADD_REMOTE_DEVICE entry will be used in that order. If the num parameter is set to zero, all other parameters are required and used if the connection is a remote device. If num is zero and the short payload (line and rel coder) is also used, it is asumed, that the local device

is addressed.

CONF_MUTE_MEDIA_CHANNEL

Tag code	NumDes	Message	SNMP Support
0xff14	no	no	no

	Datatype	Access Level	Description
Read	%	access_right_live	%
Write	t_dword	access_right_live	mutes/unmutes a media channel: Bit 1: video, Bit2: audio, Bit3: meta; session id is needed, don't mute if media channel is in multicast

CONF_STREAM_EXCLUSIVE_CHECK

Tag code	NumDes	Message	SNMP Support
0x0ba0	no - addressing via Session ID	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	check if connected ROI stream session is exclusive (meaning only one instance to one client -> exclusive ROI steering). 0: non-exclusive; 1: exclusive
Write	%	access_right_service	%

CONF_UNIT_NAME

Tag code	NumDes	Message	SNMP Support
0x0024	no	no	yes

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read the unit name
Write	p_unicode (max 32 unicode character)	access_right_service	set unit name

CONF_UNIT_ID

Tag code	NumDes	Message	SNMP Support
0x0025	no	no	yes

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read the unit ID
Write	p_unicode (max 32 unicode character)	access_right_service	set unit ID

CONF_CAMNAME

Tag code	NumDes	Message	SNMP Support
0x0019	video line	no	yes

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	Deprecated! Use CONF_CAMNAME_LINES instead.
Write	p_unicode (max 32 unicode characters)	access_right_service	Deprecated! Use CONF_CAMNAME_LINES instead.

CONF_CAMNAME2

Tag code	NumDes	Message	SNMP Support
0x0a7e	video line	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	Deprecated! Use CONF_CAMNAME_LINES instead.
Write	p_unicode (max 32 unicode characters)	access_right_service	Deprecated! Use CONF_CAMNAME_LINES instead.

CONF_CAMNAME_LINES

Tag code	NumDes	Message	SNMP Support
0x0bb1	video line	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read out camera name, use CONF_ENC_STAMPING_PROPERTIES to determine line length l (a line consists of l UTF-16 chars) and the number of supported lines n (command length = l*n |UTF-16 chars), all non used chars must be set to zero
Write	p_unicode	access_right_service	set camera name, use CONF_ENC_STAMPING_PROPERTIES to determine line length l (a line consists of l UTF-16 chars) and the number of supported lines n (command length = l*n |UTF-16 chars), all non used chars must be set to zero

CONF_HARDWARE_VERSION

Tag code	NumDes	Message	SNMP Support
0x002e	no	yes	yes

	Datatype	Access Level	Description
Read	p_string	unprotected	read the hardware version
Write	void	access_right_minimal	not supported

CONF_SOFTWARE_VERSION

Tag code	NumDes	Message	SNMP Support
0x002f	no	no	yes

	Datatype	Access Level	Description
Read	p_string	unprotected	read the software version
Write	void	access_right_minimal	not supported

CONF_SOFTWARE_VERSION_FORMATTED

Tag code	NumDes	Message	SNMP Support
0x0cd4	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the software version in the form ..
Write	void	access_right_minimal	not supported

CONF_BOOTLOADER_VERSION

Tag code	NumDes	Message	SNMP Support
0x09ef	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the bootloader version
Write	t_dword	access_right_minimal	not supported

CONF_FW_MIN_VERSION

Tag code	NumDes	Message	SNMP Support
0x0bab	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the required minimum firmware version. Format: (hexadecimal) 0xrrrrVVMM, where 'r' is reserved, 'VV' is the major version and 'MM' the minor version number. (E.g. 0x00000590 for version 5.90)
Write	t_dword	access_right_minimal	not supported

CONF_SERIAL_NUMBER

Tag code	NumDes	Message	SNMP Support
0x0ae7	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns the serial number of the device. The representation of this serial number is binary octets
Write	%	access_right_service	%

CONF_PRODUCT_NAME

Tag code	NumDes	Message	SNMP Support
0x0aea	no	no	no

	Datatype	Access Level	Description
Read	p_string	unprotected	Read the product name of the device. This command was previously named "CTN".
Write	p_string	access_right_service	not supported

CONF_COMMERCIAL_TYPE_NUMBER

Tag code	NumDes	Message	SNMP Support
0x0be7	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	Read the commercial type number (CTN) of the device. The string may be shortened or generalized as the CTN also contains parts the firmware does not know.
Write	%	access_right_service	%

CONF_DEVICE_TYPE_IDS

Tag code	NumDes	Message	SNMP Support
0x0b07	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the device type ids (hexadecimal). 1st DWORD: Product ID; 2nd DWORD: Upper byte: OEM ID (BOSCH = 0), Lower 3 bytes: Variant ID; 3rd DWORD: (Bicom) Frontend Family ID
Write	void	access_right_service	not supported

CONF_SUPPORTED_UPLOAD_TARGETS

Tag code	NumDes	Message	SNMP Support
0x0b19	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns all supported upload targets for this platform (List of n DWORDs)
Write	p_octet	access_right_service	not supported

CONF_DEVICE_GUID

Tag code	NumDes	Message	SNMP Support
0x0b22	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the device's GUID (32 bytes length)
Write	p_octet	access_right_service	write the device's GUID (32 bytes length)

CONF_DEVICE_CAPABILITIES

Tag code	NumDes	Message	SNMP Support
0x0b60	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns capabilities of the device in a tagged format
Write	p_octet	access_right_minimal	not supported

Tagged Payload Structure (T(ag) L(ength) V(alue)

16	32
num Entries 4 Bytes	
tag1 2 Byte	**length1** 2 Byte
payload1 length1 bytes	
...	
tagN 2 Byte	**lengthN** 2 Byte
payloadN lengthN bytes	

num Entries
The total number of tagged entries.

length
The length of the payload. The payload of a tag must be in network byte-order.

tags
The following tags are defined:

Tag Name	Tag ID	Tag Type	Tag description
DEVICE_TYPE_TAG	1	UInt8	1=encoder, 2=camera, 3=transcoder, 4=vrm, 5=decoder, 6=streaming gateway, 8=storage, 0=other
PTZ_CAMERA_TAG	2	UInt8	ptz capabilities yes/no; This command returns ptz capabilities if the device is a dome camera, if bilinx is supported and a bilinx camera is connected to the device or if a serial dome protocol is set (0: no ptz, 1: full ptz, 2: zoom only, no pan/tilt)
ROI_CAMERA_TAG	3	UInt8	region of interest yes/no
AUTOTRACKER_TAG	4	UInt8	autotracker yes/no
NBR_VIDEO_IN_TAG	5	UInt8	number of video inputs
NBR_TRANSCODER_TAG	6	UInt8	number of transcoders
NBR_AUDIO_IN_TAG	7	UInt8	number of audio inputs
NBR_AUDIO_OUT_TAG	8	UInt8	number of audio outputs
AUDIO_OPTIONS_TAG	9	UInt8	bit0 line in, bit1 line out, bit2 mic bit3 loadspeaker (see CONF_AUDIO_OPTIONS)
EPTZ_TRACKER_TAG	10	UInt8	eptz tracker supported or not
IMAGE_PIPE_FEATURE_TAG_OLD	11	UInt32	returns 1st 4bytes of the information that is provided by bicom command 0x0520 (ServerID 4)(for backward comaptibility). Use the new tag IMAGE_PIPE_FEATURES to get all features
BEST_FACE_TAG	12	UInt8	1 device supports CONF_BEST_FACE 0 device doesn't support CONF_BEST_FACE
PLATFORM_TYPE_TAG	13	UInt8	1-CPP3, 2-CPP4, 3-CPP-ENC, 4-CPP5, 5-CPP6, 6-CPP7, 7-CPP7.3, 255-OTHER
BICOM_DOME_TAG	14	UInt8	Indication if it is a dome that can be controlled using bicom (e.g via CONF_BICOM_COMMAND)
FPGA_TAG	15	UInt8	0 if no FPGA is available
VOUT_TAG	16	UInt8	camera has a vout (1: yes, 0: no)
HEATER_TAG	17	UInt8	camera has a heater (1: yes, 0: no)
SERIAL_PORT_TAG	18	UInt8	camera has a serial port (1: yes, 0: no)
NTCIP_TAG	19	UInt8	camera is ready for NTCIP use (1: yes - license is installed or NTCIP support comes pre-enabled, 0: no - license needs to be installed)
REPLAY_TAG	20	UInt32	supported replay option flags (DWORD) (0x01: span replay, 0x02: local replay, 0x04: transcoded replay)
GB28181_TAG	21	UInt8	supported of GB28181 (1 Byte) 0: no support 1: GB28181 is supported see CONF_GB28181 for further informations
FONT_SUPPORT_TAG	22	UInt32	supported utf-16 characters for Stamping; ASCII chars are always supported; Byte[0] full BoschSansCHS-Regular.ttf support (all chinese signs included in UTF-16) 0 not supported 1 supported; Byte[1-3] reserved
			record format version (DWORD, offset 0), record format version minor (WORD, offset 4), recorder version (WORD, offset

RECORDER_TAG	23	RecorderVersion	6) supported recorder mode flags (DWORD, offset 8) (0x01: standard recording, 0x02: dual recording, 0x04: buffered recording)
PTZ_ON_CLIENT_TAG	24	UInt8	0 not necessary 1: dewarping on client makes sense and all required data is provided via SEI information in the h.264 stream. Normally the case on panoramic cameras with a large view angle 180° cameras.
VIRTUAL_LINES_TAG	25	UInt8	Returns the number of virtual lines on this device or zero if there are none. If there are virtual lines, more information is available via CONF_VIRTUAL_LINES.
DPTZ_STATIC_CAPS_TAG	26	PUInt8	Signal PTZ capabilities see CONF_DPTZ_STATIC_CAPS for a detailed description of the payload
MAX_DEC_RESOLUTION_TAG	27	Resolution	max decoder/transcoder input width (DWORD, offset 0), max decoder/transcoder input height (DWORD, offset 4)
MAX_SUBVIEWS_TAG	28	UInt8	max number of video subviews
FIRE_DETECTION_TAG	29	UInt8	Fire detection available (1: yes, 0: no)
FW_UPLOAD_SIGNATURE_TAG	30	UInt8	Firmware upload signature required (1: only signed firmware accepted, 0: unsigned firmware accepted)
IMAGE_PIPE_FEATURE_TAG	31	PUInt8	returns the information that is provided by bicom command 0x0520 (ServerID 4)
CLUSTERED_DEVICE_TAG	32	UInt8	indication if the camera is clustered with other devices (1: yes, 0: no)
PIR_TAG	33	UInt8	number of PIR (passive infrared) sensors
AMBIENT_LIGHT_DETECT_TAG	34	UInt8	ambient light detection (0: no, 1: yes)
SUPPORT_SUDO_URL	35	UInt8	/sudo URL is available to switch users (0 or not present: no, 1: yes)
IS_IN_FORCE_PWD_MODE	36	UInt8	Device has no password at the moment and will force the user to set one
IS_PTZ_IN_VCD	37	UInt8	PTZ information can be found in VCD stream (this capabilty is necessary for the VCA global field feature)
CODER_SPEC_ENC_PROF	38	UInt8	coder specific encoder presets
POE_BASE_CLASS_TAG	39	UInt8	base PoE class of device, without consideration of power adder values (1: class0, 2: class1, 3: class2, etc. 255: no PoE)
SOD_DETECTION_TAG	40	UInt8	SOD detection is available
VIDEO_LINE_DUO	41	UInt8	device signals duo line possibility (bit per camera line)
ISCSI_SUPPORT	42	UInt32	iSCSI capabilities (bitfield, each bit represents a capability):\n\tBit 0: Support for NetApp E2800\n\tBit 1 : Support for Multipathing(\"NetApp specific style\")
INDIVIDUAL_ENCODER_OPERATION_MODE_CONFIG	43	UInt16	1: invidual encoder configuration mode: In CONF_CODER_VIDEO_OPERATION_MODE, num parameter for a specific absolute coder has to be provided; global configration (numDesc = 0) is not supported 0: global encoder configuration mode : in CONF_CODER_VIDEO_OPERATION_MODE, num parameter supports only 0 as global option for all possible encoders
WIFI_TAG	44	UInt8	Wifi support (0: no, 1: yes, as station, 2: yes, as access point)
C_MOUNT_LENS_TYPE	45	UInt8	Device has a changable lens C-mount (0: device has a fix attached lens, 1: yes, lens is changable)
TC_ENC_H265_TAG	46	UInt8	
NET_DYNAMIC	47	UInt8	device supports network configuration at runtime
KINESIS_SUPPORT	48	UInt8	Decice supports streaming to kinesis
AWS_S3_SUPPORT	49	UInt8	Device supports streaming to amazon s3
RCP_UDP_PAYLOAD_ENCRYPTION	50	UInt8	Device supports rcp udp payload encryption
IMU_TAG	51	UInt8	number of IMU sensors (Inertial Measurement Unit; acceleration, gyroscope, ..)
CLOUD_WATCH_SUPPORT	52	UInt8	device supports logging to cloud watch
CALIBRATION_LINE_DEPENDENCIES	53	PUInt8	returns the information calibration dependencies between lines, if missing then all lines are independent,

CONF_DEVICE_OPERATION_MODE_VERSIONS

Tag code	NumDes	Message	SNMP Support
0x0b61	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	information about device operation_modes, see
Write	p_octet	access_right_minimal	not supported

Response Payload Structure

operation mode version info 1 X[1] Bytes ...	
N x X[i] Bytes	**operation mode version info N** X[N] Bytes ...

operation mode version info 1 - N

A list of operation mode version info structures.

type	specific information
1 Byte	N Bytes ...

type

type of the operation mode version info structure.

Values:

recording configuration version	0x01

specific information

specific information depending on the type.

operation mode version info: recording configuration version structure

0x01	variant	version
1 Byte	1 Byte	2 Bytes

variant

Varian of the recording configuration.

Values:

profile configuration mode	0	Configuration has to be done by command CONF_HD_RECORD_PROFILES
stream configuration mode	1	Configuration has to be done by command CONF_HD_RECORD_PROFILES_V2

version

Version of the recording configuration. Current version is 1.

CONF_DEVICE_OPERATION_MODE_VERSIONS

This command can be used to get a hint for changes between firmware versions or differences between device types concerning handling or way of configuration of the devices for several aspects. In order to parse the response payload, first check the first byte of a operation mode version info structure, which is the type and depending on the type read the remaining payload of the structure. Do this again for all following operation mode version info structures until the end of the response payload.

CONF_SOFT_VARIANT_ID

Tag code	NumDes	Message	SNMP Support
0x0bb2	-	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the soft variant Id if set; (notice: if no soft variant ID is set, the 'original' variantID will be returned)
Write	t_dword	access_right_service	Set a soft variant ID. Only possible on certain devices. For a list of allowed soft variants see CONF_SOFT_VARIANT_ID_OPTIONS. All configuration except IP/subnet mask and DHCP settings will be set to default and the device reboots automatically. (Notice: only for some special soft variants which have the option flag 'no factory defaults performed' (see CONF_SOFT_VARIANT_ID_OPTIONS), the device configuration will be kept and not set to defaults.

CONF_SOFT_VARIANT_ID_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bb6	-	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get a list of possible soft variant IDs plus the according variant name. 1st byte: number of IDs. 2nd byte: Flags (0x01: no 'factory defaults' performed). 3rd-4th byte reserved. Then n times: 1 dword id + 60 bytes (zero-terminated) camera name string; Notice: if no soft variant IDs are possible, an empty payload will be returned.
Write	p_octet	access_right_service	not supported

CONF_APPLICATION_TYPE

Tag code	NumDes	Message	SNMP Support
0x0c47	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	returns the application type
Write	t_octet	access_right_minimal	-

CONF_SECURITY_COPROC_VERSION

Tag code	NumDes	Message	SNMP Support
0x0b93	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the security coproc version (read-only)
Write	%	access_right_minimal	%

CONF_SECURITY_COPROC_CERTIFICATE

Tag code	NumDes	Message	SNMP Support
0x0b91	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the security coproc device certificate (X.509-certificate ASN.1-DER coded)
Write	%	access_right_minimal	%

CONF_SECURITY_COPROC_AUTHENTICATE

Tag code	NumDes	Message	SNMP Support
0x0b92	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read with incoming payload: send a challenge for device authentication (16 bytes: 8 random bytes + the 8 byte sequence 0x4E, 0x58, 0x50, 0x48, 0x41, 0x2E, 0x30, 0x31), read the 128 bytes challenge response
Write	%	access_right_minimal	%

CONF_TRANSCODER_INFORMATION

Tag code	NumDes	Message	SNMP Support
0xd060	-	no	no

	Datatype	Access Level	Description
Read	-	access_right_live	see detailed description
Write	%	access_right_minimal	%

Payload Structure

sequence of

16	32
tag 2 Byte	**len** 2 Byte
len bytes data	

tag

Values:

Transcoder state: 0: Not present, 1: connected	1
Version (null terminated string)	2
Number of sessions total	3
Number of sessions local	4
Number of sessions dedicated	5
Sessions in use total	6
Sessions in use local	7
Sessions in use dedicated	8
Session in use local by other VRM	9
Sessions available total	10
Sessions available local	11
Sessions available dedicated	12
Sessions offline	13

len

len of data

data

len bytes of data

CONF_TRANSCODER_CAPABILITIES

Tag code	NumDes	Message	SNMP Support
0x0c0d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns capabilities of a transcoder device in a tagged format
Write	p_octet	access_right_minimal	not supported

lists transcoder capabilities in a tagged format.

Tagged Payload Structure (T(ag) L(ength) V(alue)

16 32

num Entries 4 Bytes	
tag1 2 Byte	**length1** 2 Byte
payload1 length1 bytes	
...	
tagN 2 Byte	**lengthN** 2 Byte
payloadN lengthN bytes	

num Entries

The total number of tagged entries.

length

The length of the payload. The payload of a tag must be in network byte-order.

tags

The following tags are defined:

Tag Name	Tag ID	Tag Type	Tag description
TC_TAG_MAX_RESOLUTION	0	Resolution	
TC_TAG_MAX_SUBVIEWS	0	UInt8	

CONF_HOST_NAME

Tag code	NumDes	Message	SNMP Support
0x0cda	no	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read the host name
Write	p_unicode (max 128 unicode character)	access_right_service	set host name

CONF_STARTPAGE_BACKGROUND_URL

Tag code	NumDes	Message	SNMP Support
0x028d	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the URL of the startpage' background image (max 64 char)
Write	p_string	access_right_service	set the URL of the startpage' background image (max 64 char)

CONF_STARTPAGE_LOGO_URL

Tag code	NumDes	Message	SNMP Support
0x028e	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the URL of the startpage' OEM logo image (max 64 char)
Write	p_string	access_right_service	set the URL of the startpage' OEM logo image (max 64 char)

CONF_STARTPAGE_PRESENTATION_SWITCHES

Tag code	NumDes	Message	SNMP Support
0x028f	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get switches for HTML startpage presentation (usage is up the the user)
Write	t_dword	access_right_service	set switches for HTML startpage presentation (usage is up the the user)

CONF_DYNAMIC_HTML_NAME

Tag code	NumDes	Message	SNMP Support
0x0298	yes	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the name for a dynamic HTML page (num range 1..4)
Write	p_string	access_right_service	set the name for a dynamic HTML page (num range 1..4) (max. 19 char)

CONF_DYNAMIC_HTML_DATA

Tag code	NumDes	Message	SNMP Support
0x0299	yes	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	set the data(content) for a dynamic HTML page (num range 1..4) (max. 2048 char)
Write	p_octet	access_right_service	set the data(content) for a dynamic HTML page (num range 1..4) (max. 2048 char)

CONF_BROWSER_LANGUAGE_VAL

Tag code	NumDes	Message	SNMP Support
0x01e3	no	no	no

	Datatype	Access Level	Description
Read	t_octet	unprotected	read current set browser language
Write	t_octet	access_right_service	set browser language (numbering depends on toolkit implementation)

CONF_BROWSER_DATETIME_FORMAT_VAL

Tag code	NumDes	Message	SNMP Support
0x01e9	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read browser date/time format
Write	t_octet	access_right_service	set browser date/time format (1=Europe, 2=USA, 3=Japan)

CONF_DATE_WDAY

Tag code	NumDes	Message	SNMP Support
0x0027	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	"Sunday" ... "Saturday"; read the weekday according to the systems date setting
Write	void	access_right_service	not supported

CONF_DATE_DAY

Tag code	NumDes	Message	SNMP Support
0x0028	no	no	no

	Datatype	Access Level	Description
Read	t_octet or p_string (2 chars length)	access_right_minimal	read the day of month
Write	t_octet	access_right_service	set the day of month, not supported while recording is running or configured to active

CONF_DATE_MONTH

Tag code	NumDes	Message	SNMP Support
0x0029	no	no	no

	Datatype	Access Level	Description
Read	t_octet and p_string (2 chars length)	access_right_minimal	read the month
Write	t_octet	access_right_service	set the month, not supported while recording is running or configured to active

CONF_DATE_YEAR

Tag code	NumDes	Message	SNMP Support
0x002a	no	no	no

	Datatype	Access Level	Description
Read	t_word and p_string (4 chars length)	access_right_minimal	read the year
Write	t_word	access_right_service	xx set the year; century needed, not supported while recording is running or configured to active

CONF_TIME_HRS

Tag code	NumDes	Message	SNMP Support
0x002d	no	no	no

	Datatype	Access Level	Description
Read	t_octet and p_string (2 chars length)	access_right_minimal	read the hours
Write	t_octet	access_right_service	set the hours, not supported while recording is running or configured to active

CONF_TIME_MIN

Tag code	NumDes	Message	SNMP Support
0x002c	no	no	no

	Datatype	Access Level	Description
Read	t_octet and p_string (2 chars length)	access_right_minimal	read the minutes
Write	t_octet	access_right_service	set the minutes, not supported while recording is running or configured to active

CONF_TIME_SEC

Tag code	NumDes	Message	SNMP Support
0x002b		no	no

	Datatype	Access Level	Description
Read	t_octet and p_string (2 chars length)	access_right_minimal	read the seconds
Write	t_octet	access_right_service	set the seconds, not supported while recording is running or configured to active

CONF_TIMEZONE

Tag code	NumDes	Message	SNMP Support
0x024e	no	no	yes

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	the timezone in which the unit has to operate (GMT +- nbr of seconds +- nbr of seconds DLS), read only version 2.50
Write	t_int	access_right_service	the timezone in which the unit has to operate (GMT +- nbr of seconds +- nbr of seconds DLS), read only since version 2.50

CONF_UTC_ZONEOFFSET

Tag code	NumDes	Message	SNMP Support
0x031f	no	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	the timezone in which the unit has to operate (GMT +- nbr of seconds)
Write	t_int	access_right_service	the timezone in which the unit has to operate (GMT +- nbr of seconds)

CONF_NTP_SERVER_IP_STR

Tag code	NumDes	Message	SNMP Support
0x024f	no	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	reads the time servers ip address ("xxx.xxx.xxx.xxx", or URL) used to sync time from
Write	p_string	access_right_service	sets the ntp server ip (as string; "xxx.xxx.xxx.xxx" or URL) to sync time from

CONF_NTP_SYNC_MODE

Tag code	NumDes	Message	SNMP Support
0x031e	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns mode which is used by time server, time server(0), SNTP server(1), off(2)
Write	t_dword	access_right_service	selects mode which is used to sync time on this machine, time server(0) SNTP server(1), off(2), tls_date(3)

CONF_NTP_SYNC_MODE_INTERNAL

Tag code	NumDes	Message	SNMP Support
0x0321	Message : no	Description : returns mode which is used by time server, time server(0), SNTP server(1), off(2)	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	t_dword	access_right_internal	selects mode which is used to sync time on this machine, time server(0) SNTP server(1), off(2), tls_date(3), writing does not implicitly clear CONF_ALLOW_OVERWRITE_TIMESRVIP_BY_DHCP

CONF_DAY_LIGHT_SAVE_TIME_TABLE

Tag code	NumDes	Message	SNMP Support
0x0987	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read 20 entries for daylight save time: UTC time in sec since 2000 (4byte) and associated offset in sec (4 signed bytes), unused entries have to be filled with zero
Write	p_octet	access_right_service	write 20 entries for daylight save time: UTC time in sec since 2000 (4byte) and associated offset in sec (4 signed bytes), unused entries have to be filled with zero

CONF_DAY_LIGHT_SAVE_TIME

Tag code	NumDes	Message	SNMP Support
0x0988	no	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	read the currently used day light save time offset in seconds
Write	t_int	access_right_service	write the day light save time offset in seconds; the device updates this from DAY_LIGHT_SAVE_TIME_TABLE to show current offset used

CONF_FORCE_TIME_SET

Tag code	NumDes	Message	SNMP Support
0x0a0f	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the time, 8 bytes payload, offset 0: year (word); offset 2: month (octet); offset 3: day (octet); offset 4: hrs (octet); offset 5: min (octet); offset 6: sec (octet); offset 7: reserved (octet)
Write	p_octet	access_right_service	set the time, if recording is running it will be stopped and restarted, 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)

CONF_SYSTEM_DATETIME_V2

Tag code	NumDes	Message	SNMP Support
0x0ba8	if set to 0xFFFF the daylight saving values are not overwritten if not present	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	gets the system date and time and the daylight savings in one step: yyyy-mm-ddThh:mm:ss; PosixTimezone; optional string: yyyy->year,mm->month,dd->day,T->beginning of the time section,hh->hour,mm->minute,ss->second (all time values are utc time); The Time Zone format is specified by POSIX, refer to POSIX 1003.1 section 8.3. (the PosixTimezone string is only included if written by that command. If not present the corresponding settings are cleared).
Write	p_string	access_right_service	writes the system date and time and the daylight savings in one step: yyyy-mm-ddThh:mm:ss; PosixTimezone; optional string: The Time Zone format is specified by POSIX, refer to POSIX 1003.1 section 8.3

CONF_ALLOW_OVERWRITE_TIMESRVIP_BY_DHCP

Tag code	NumDes	Message	SNMP Support
0x0c0e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	allow or deny that dhcp overwrites time/ntp srv ip, 0 = deny, 1 = allow
Write	t_octet	access_right_service	allow or deny that dhcp overwrites time/ntp srv ip, 0 = deny, 1 = allow

CONF_OPTIONAL_TIME_SERVER_PORT

Tag code	NumDes	Message	SNMP Support
0x0c40	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the optional/alternative time server port. If not set (0), the according standard port will be used (depending on the configured time server mode)
Write	t_word	access_right_service	Set an optional/alternative time server port. If not set (0), the according standard port will be used (depending on the configured time server mode).

CONF_NTP_START_SERVER

Tag code	NumDes	Message	SNMP Support
0x0c63	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	reads if the NTP server of the camera should be started
Write	f_flag	access_right_service	set if the NTP server of the camera should be started (0 = do not start; 1 = start server)

CONF_TIME_SC_CONNECT_FAIL_MSG

Tag code	NumDes	Message	SNMP Support
0x0ce0	no	yes	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_internal	neither read nor write access, only message, 1 = connect to server for time sync failed
Write	t_octet	access_right_service	neither read nor write access, only message, 1 = connect to server for time sync failed

CONF_PASSWORD_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x028b	password level	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the unit password (scrambled). num parameter sets the password levels; 1=user, 2=service, 3=live
Write	p_string	access_right_service	set the unit password. num parameter sets the password levels; 1=user, 2=service, 3=live; Notice: only standard ASCII characters allowed!

CONF_REMOTE_PASSWORD

Tag code	NumDes	Message	SNMP Support
0x010c	destination IP number	no	no

	Datatype	Access Level	Description
Read	void	access_right_minimal	not supported
Write	p_string	access_right_service	deposit the password of the called station

CONF_ACCOUNT_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x0b5e	yes (index of the account 1...4)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	gets the account information: 4 bytes types (0: no; 1: ftp; 2: dropbox),32 bytes: name, 64 bytes username (ftp only), 32 bytes password (ftp only), 64 bytes path, 64 bytes optional url (e.g ip of the ftp server, unused for dropbox)
Write	p_octet	access_right_service	stores the account information: 4 bytes types (0: no; 1: ftp; 2: dropbox; 4: recording), 32 bytes name, 64 bytes username (ftp only), 32 bytes password (ftp only), 64 bytes path, 64 bytes optional url (e.g ip of the ftp server, unused for dropbox), 4 bytes optional flags (e.g. FTP Encryption mode 0=off,1=TLS)

CONF_ACCOUNT_SETTINGS_V2

Tag code	NumDes	Message	SNMP Support
0x0cc0	yes (index of the account 1...4)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns account settings in a tagged format
Write	p_octet	access_right_service	....

lists account tags.

Tagged Payload Structure (T(ag) L(ength) V(alue)

16	32
num Entries 4 Bytes	
tag1 2 Byte	**length1** 2 Byte
payload1 length1 bytes	
...	
tagN 2 Byte	**lengthN** 2 Byte
payloadN lengthN bytes	

num Entries
The total number of tagged entries.

length
The length of the payload. The payload of a tag must be in network byte-order.

tags
The following tags are defined:

Tag Name	Tag ID	Tag Type	Tag description
ACCOUNT_TAG_TYPE	1	UInt32	0: not configured, 1: ftp, 2: dropbox, 5: amazon s3, 6: kinesis
ACCOUNT_TAG_NAME	2	PChar	account name
ACCOUNT_TAG_URL	3	PChar	url, e.g url to the ftp server
ACCOUNT_TAG_LOGIN	4	PChar	username
ACCOUNT_TAG_PASSWORD	5	PChar	password
ACCOUNT_TAG_PATH	6	PChar	path
ACCOUNT_TAG_FLAGS	7	UInt32	optional flags (e.g. FTP Encryption mode 0=off,1=TLS)
ACCOUNT_TAG_ACCESS_KEY	8	PChar	aws access key (max 128 bytes)
ACCOUNT_TAG_SECRET_KEY	9	PChar	aws secret access key (max 128 bytes)
ACCOUNT_TAG_BUCKET_NAME	10	PChar	bucket name (amazon s3), max 128 bytes
ACCOUNT_TAG_REGION	11	PChar	aws region
ACCOUNT_TAG_CAMERA_ID	12	PChar	s3 camera id, string will be added to the uploaded filename (max 128 bytes)
ACCOUNT_TAG_FILE_DURATION	13	PChar	s3 file duration in seconds
ACCOUNT_TAG_STREAM_NAME	14	PChar	stream name
ACCOUNT_TAG_KBPS	15	UInt32	maximum datarate for the backup, see CONF_BACKUP_MAX_KBPS
ACCOUNT_TAG_GROUP	16	PChar	cloud watch log group
ACCOUNT_TAG_CLOUD_WATCH_ENABLED	17	UInt8	cloud watch enabled/disabled

CONF_NBR_OF_ACCOUNTS

Tag code	NumDes	Message	SNMP Support
0x0b62	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the max number of accounts
Write	t_dword	access_right_service	not supported

CONF_DROPBOX_AUTH_ADDR

Tag code	NumDes	Message	SNMP Support
0x0b81	account idx	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	gets the url that should be used to login at dropbox (payload url)
Write	p_string	access_right_minimal	gets the url that should be used to login at dropbox (payload url)

CONF_DROPBOX_AUTH_STATUS

Tag code	NumDes	Message	SNMP Support
0x0b82	account idx	yes	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	gets authentication status (0: not authenticated, 1: authenticated, 2: authentication session timed out, 3: failed)
Write	t_octet	access_right_minimal	not supported

CONF_DROPBOX_TOKEN_V2

Tag code	NumDes	Message	SNMP Support
0x0c7a	account idx	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	gets the dropbox token (v2)
Write	p_octet	access_right_service	sets the dropbox token (v2), max 256 bytes

CONF_DROPBOX_AUTH_V2

Tag code	NumDes	Message	SNMP Support
0x0c7b	account idx	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	not supported
Write	p_string	access_right_service	the dropbox authorization code and start the process to get the dropbox access token

CONF_ACCOUNT_STATUS

Tag code	NumDes	Message	SNMP Support
0x0bde		message only	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	provides information if connection to a account fails (1: ftp, 2: dropbox)
Write	t_octet	access_right_minimal	provides information if connection to a account fails (1: ftp, 2: dropbox)

CONF_ADV_USER_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x0bda	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	
Write	p_octet	access_right_service	

Payload Structure

16	32
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
8	24

length

length of tagged value including length and tag field

tag

tag specifying the encoding and meaning of the value

value

value payload data (length - header bytes), can contain tagged values

General Description

Command CONF_ADV_USER_SETTINGS allows to query or configure user entries for camera access control. User entries are indentified by the user name, the user name of each entry must be unique.
In read direction only the user name is used to find the user entry to query the user configuration.
In write direction an already existing user with the same name is modified or a new user with the given configuration is created.
Supported lengths, types and restrictions can be queried using CONF_ADV_USER_SETTINGS_OPTIONS.

Tag Structure

16	32
length 2 Byte	**tag = 0 (User Entry)** 2 Byte
length 2 Byte	**tag = 1 (User Name)** 2 Byte
user name (length - header) Bytes ...	
length 2 Byte	**tag = 2 (User Type)** 2 Byte
user type 4 Bytes (network byte order)	
length 2 Byte	**tag = 3 (User Group)** 2 Byte
user level 4 Bytes (network byte order)	
length 2 Byte	**tag = 4 (User Password)** 2 Byte
user password (length - header) Bytes ...	
length 2 Byte	**tag = 5 (Expires)** 2 Byte
Seconds this user should live (only when type == 3) 4 Bytes (network byte order)	
8	24

tag = 0 (User Entry)

The payload of tag user entry contains further tagged values described below.

tag = 1 (User Name)

The payload of tag user name contains the user name string which is used to identify a user.

tag = 2 (User Type)

The payload of tag user type contains the user type (values defined below).
Changing the type of an existing user to Deleted will delete the user.
A User Entry of type Password must contain a password entry. A User Entry of type 'Temporary with password' must contain a password entry and an 'Expires' entry.

user type

Values:

Deleted	0
Password	1
Certificate	2

Temporary with password	3
Cloud	4 (read only)

tag = 3 (User Group)

The payload of tag user group contains the user group. Each group has a unique ID you can request the group name with a seperate RCP command.

user groups

Values:

Live Group	1
User Group	2
Service Group	3

tag = 4 (User Password)

The payload of tag user password contains the user password.
This field is only mandatory when configuring a user of type Password and will be empty when read if there is no password or just '********' when a password is set.

tag = 5 (Expires)

The 'Expires' indicates the life time of the user in seconds. So if the user should life for 1 hour pass the value 3600 here.
The field is mandatory when you create a user of type 'Temporary with password' (3). For all other types this field should not be present or have the payload 0.
On read, this field will indicate the remaining time in seconds until the user will expire.

CONF_ADV_USER_SETTINGS_LIST

Tag code	NumDes	Message	SNMP Support
0x0bdc	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	returns a list of tagged values (User Entries), see CONF_ADV_USER_SETTINGS for a detailed desciption of the payload format
Write	p_octet	access_right_service	not supported

CONF_ADV_USER_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bd9	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	4 bytes max number of users, 4 bytes max user name length, 4 bytes max password length, 1 byte password user support [0 = no, 1 = yes], 1 byte certificate user support [0 = no, 1 = yes]
Write	p_octet	access_right_minimal	not supported

CONF_USER_AUTH_MODE

Tag code	NumDes	Message	SNMP Support
0x0be3	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	
Write	p_octet	access_right_service	

General Description

Command CONF_USER_AUTH_MODE allows to query or enable/disable user authentication modes.

Payload Structure

16	32
length 2 Byte	**tag = 0 (Auth Modes)** 2 Byte
active user auth modes 32 Bits	

8 24

length

length of tagged value including length and tag field

tag = 0 (Auth Modes)

tag specifying the encoding and meaning of the value

active user auth modes

bit mask containing the active user authentication modes

Values:

Password Auth	1
Certificate Auth	2
Active Directory (ADFS) Auth	4

CONF_MIN_TLS_VERSION

Tag code	NumDes	Message	SNMP Support
0x0c56	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	Get the TLS version minimal required to connect to the device. Word coding: bit 15-8 mayor version, bit 7-0 minor version. E.g. 0x0102 is TLSv1.2
Write	t_word	access_right_service	Set the TLS version minimal required to connect to the device. Word coding: bit 15-8 mayor version, bit 7-0 minor version. E.g. 0x0102 is TLSv1.2

CONF_ENABLE_MASTERPWD

Tag code	NumDes	Message	SNMP Support
0x0c6d	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	Shows if the the password reset mechanism is enabled (== 1) or disabled (== 0).
Write	f_flag	access_right_service	Enable or disable the password reset mechanism (1 == enable, 0 == disable). Disable it if you don't want that Bosch support can reset the passwords of your camera.

CONF_ACCESS_OPTION

Tag code	NumDes	Message	SNMP Support
0x0c7f	no	no	no

	Datatype	Access Level	Description
Read	t_octet	unprotected	read the current access options. 0=normal access, Bit1: camera is in 'force initial password' mode. I.e. before further usage an initial service password must be set.
Write	t_octet	access_right_service	not supported

CONF_MASTERPWD_CHALLENGE

Tag code	NumDes	Message	SNMP Support
0x013f	no	no	no

	Datatype	Access Level	Description
Read	p_string	unprotected	generates an password challenge which has to be passed to the operator for password reset
Write	p_string	unprotected	provide the generated unlock key from operator to reset passwords. The key has to be in hex string repesentation without 0x in front.

CONF_MASTERPWD_CHALLENGE_STATE

Tag code	NumDes	Message	SNMP Support
0x0c21	no	no	no

	Datatype	Access Level	Description
Read	p_octet	unprotected	retrieves status information about the masterpwd challenge (4 bytes validity in seconds, 4 bytes lock state)
Write	%	unprotected	%

CONF_CERTIFICATE

Tag code	NumDes	Message	SNMP Support
0x0be9	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

16	32
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	

8 24

Generic description

The command CONF_CERTIFICATE supports the following operations:

- Upload a certificate optionally including its private key.
- Download a certificate or signing request from the device.
- Delete a certificate or signing request. A delete implicitly deletes an associated private key.

Each certificate/signing request/key is identified by a unique label.
If a certificate has a matching private key both have the same label.
When the command fails it returns a command specific error code in the payload
On upload the command supports the following file formats:

- PEM: PEM file format. With this format you can upload a certificate and a key at once. For this you must set Certificate and Private Key bits in type tag.
- DER: Binary DER file format. For this file format you can only upload one type of content and the type must be exactly specified in type tag.
- PKCS#12 or PFX: PKCS#12 (sometimes also called PFX) file format. This file format carried information what is included, so you can upload certificate and key in one file. The type tag does not matter when using this file format and can be set to certificate and key or even be not provided

You can also use this command to upload encrypted private keys as PEM, DER or PKCS12 file. To do this you upload the file containing the encrypted key. The device stores the encrypted key internally and waits, that the password is provided later with help of the CONF_CERTIFICATE_REQUEST command.
To **delete** a certificate/CSR or key you have to provide the following tags: Tag 0 with the label, tag 1 with 0, tag 2 with length 4 and no data and tag 3 with entry type 0xFF (delete).
Example Payload for delete of cert with label 'test': 0008000074657374 0008000100000000 00040003000000FF

Tag Label

Contains the label string, no zero termination. For the maximum supported length see CONF_CERTIFICATE_OPTIONS.

length 2 Byte	**tag = 0** 2 Byte
certificate/signing request/key label (length - header) Bytes...	

16 32

Tag Format

Specifies the certificate/signing request/key data format (PEM / DER / PKCS#12(PFX) is supported).

length = 8 2 Byte	**tag = 1** 2 Byte
format 4 Bytes	

16 32

Entry type

Values:

PEM	0
DER	1
PKCS#12 / PFX	2

Tag Data

Contains the certificate/signing request/key data in the specified format.
If this tag contains no data (zero payload length) the corresponding entry is deleted from the device.

length 2 Byte	**tag = 2** 2 Byte
data (length - header) Bytes...	

16 32

Tag Entry Type

Specifies the entry type for read and write. This tag is mandatory for PEM and DER format and should appear only once.

length = 8 2 Byte	**tag = 3** 2 Byte
Entry type 4 Bytes	

16 32

Entry type

Values:

Certificate	1
Signing request(read or delete only)	2
Private key(write only)	4
Encrypted key(write only)	8
Any type(delete only)	255

Tag Protection

Contains a bit mask showing the protection bits of the certificate. Each bit of the 32 bit value shows a specific protection.

length = 8 2 Byte	**tag = 4** 2 Byte
Protection Bits 4 Bytes	

16 32

Protection bits

Values:

Protect at factory reset	bit0(0x1)

Specific Error Types

CERTIFICATE_CMD_ERROR_FAIL	0x00
CERTIFICATE_CMD_ERROR_INV_ARGS	0x01
CERTIFICATE_CMD_ERROR_ALREADY_EXISTS	0x02
CERTIFICATE_CMD_ERROR_FORMAT	0x03
CERTIFICATE_CMD_ERROR_NO_CERT_FOR_KEY	0x04
CERTIFICATE_CMD_ERROR_NO_KEY_ENTRY	0x05
CERTIFICATE_CMD_ERROR_NO_STORAGE_SPACE	0x06

CONF_CERTIFICATE_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bea	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	4 bytes max label length, 1 byte certificate store support [0 = no, 1 = yes]
Write	p_octet	access_right_minimal	not supported

CONF_CERTIFICATE_LIST

Tag code	NumDes	Message	SNMP Support
0x0beb	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
8	24

length

length of tagged value including length and tag field

tag

tag specifying the encoding and meaning of the value

value

value payload data (length - header bytes), can contain tagged values

Generic description

Command CONF_CERTIFICATE_LIST allows to query the labels of all certificates, signing requests and keys installed on a device.
The command CONF_CERTIFICATE can then be used to retrieve a certificate or signing request using the label.
The format has an internal tagged structure. Some tags are optional and might be absent if the data is not available. The tags are always in the sequence as listed below.

Tag Structure

16	32
length 2 Byte	**tag = 0 (List entry)** 2 Byte
length 2 Byte	**tag = 1 (Entry label)** 2 Byte
entry label (length - header) Bytes ...	
length 2 Byte	**tag = 2 (Type)** 2 Byte
type 4 Bytes	
length 2 Byte	**tag = 3 (Subject Common name)** 2 Byte
certificate subject common name (length - header) Bytes ...	
length 2 Byte	**tag = 4 (Protection Bitmap)** 2 Byte
Protection Bits of this entry 4 Bytes ...	
length 2 Byte	**tag = 5 (Issuer Common name)** 2 Byte
certificate issuer common name (length - header) Bytes ...	
length 2 Byte	**tag = 6 (NotAfterUTC)** 2 Byte
(optional see below) certificate not after as UTC string (length - header) Bytes ...	
length 2 Byte	**tag = 7 (NotAfterGT)** 2 Byte
(optional)certificate not after as General Time (GT) string (length - header) Bytes ...	
8	24

tag = 0 (List entry)

The payload of tag list entry contains further tagged values described below.
For each certificate, signing request or key with unique label on the device one list entry tag is contained in the command payload.

tag = 1 (Entry label)

Contains the entry label string, no zero termination.

tag = 2 (Entry type)

Contains the entry type bitmask. All types with the same label are listed together as one entry.

entry type

Values:

Certificate	bit0(0x1)
Signing request	bit1(0x2)
Private key	bit2(0x4)
Encrypted private key	bit3(0x8)
Certificate generation in progress	bit4(0x10)
Signing request generation in progress	bit5(0x20)
Key decryption in progress	bit6(0x40)
Encrypted PKCS#12	bit7(0x80)

tag = 3 (Certificate's subject common name)

Contains the subject common name string (UTF-16) of the certificate or signing request. This entry is optional and only used for certificates and signing requests.

tag = 4 (Protection Bitmap)

Show the protection bits of this entry. This tag is optional if the type supports protection bits and if any protection is set. If this tag is missing, the item is unprotected. The following values are possible:

Protection Types

Values:

Protect at factory reset	bit0(0x1)
Protect from manual deletion	bit1(0x2)

tag = 5 (Certificate's issuer common name)

Contains the issuer common name string (UTF-16) of the certificate. This entry is optional and only used for certificates.

tag = 6 (Certificate's not after timestamp in UTC)

Contains the UTC string (UTF-16, format YYMMDDhhmmssZ) of the not after timestamp of the certificate . This entry is optional and only used for certificates. If tag = 7 (Certificate's not after timestamp in GT) is present you should always use the date from there as UTC is limited to 1950-01-01 up to 2049-12-31. For compatibility reason both tags will be present when the date can be presented in both formats.

tag = 7 (Certificate's not after timestamp in GT)

Contains the General time (GT) string (UTF-11, format YYYYMMDDhhmmssZ) of the not after timestamp of the certificate. This entry is optional and only used for certificates. Thisa entry should always be used instead of tag 6 (Certificate's not after timestamp in UTC) because of it's wider time range.

CONF_CERTIFICATE_REQUEST

Tag code	NumDes	Message	SNMP Support
0x0bec	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	not supported
Write	p_octet	access_right_service	see detailed description

Payload Structure

length 2 Byte	tag 2 Byte
payload (length - header) Bytes	

16 32

Generic description

The command CONF_CERTIFICATE_REQUEST allows to start a certificate signing request or a key decryption operation.
Since both operations may take some time to complete, this command may return before the actual completion of the operation.
The progress of the operation can be tracked by registering for the CONF_CERTIFICATE_REQUEST_PROGRESS message.

Tag Label

Contains the label string, no zero termination. Access the generated private key and the signed certificate using the provided label after the process has finished successfully.
For key decryption this label is used to identify the key to be decrypted.

length 2 Byte	tag = 0 2 Byte
certificate/key label (length - header) Bytes...	

16 32

Tag Type

Specifies the certificate operation type.
The mandatory/supported tags of the command vary based on the operation type. See CONF_CERTIFICATE_REQUEST_OPTIONS for a list of supported commands.

length 2 Byte	tag = 2 2 Byte
Type 4 Bytes	

16 32

Type

Values:

create PKCS#10 CSR	0
create self-signed certificate	3
decrypt private key	4

Tag Key Type

Specifies the key type for the signing request/certificate to be generated. The list of supported types for a specific device can be requested using command CONF_CERTIFICATE_REQUEST_OPTIONS.

length = 4 2 Byte	tag = 1 2 Byte
key type 4 Bytes	

16 32

key type

Below enumeration lists example values for the key type association:

Values:

RSA 1024bit	0
RSA 2048bit	1
Elliptic Curve P256	2

Tag CA Server IP

Specifies the IP address string of the CA server, no zero termination.
This tag is optional and only used when a signing request is sent to a server.

length 2 Byte	tag = 3 2 Byte
server IP (length - header) Bytes...	

16 32

Tag CA Server Port

CA server port: tag = 4

Specifies the CA server port.
This tag is optional and only used when a signing request is sent to a server.

length 2 Byte	**tag = 4** 2 Byte
server port 4 Bytes	

16 32

Tag Common Name

Contains the common name string (UTF-16) for the certificate/signing request, no zero termination. This tag is optional.

length 2 Byte	**tag = 5** 2 Byte
common name (length - header) Bytes...	

16 32

Tag Organization Name

Contains the organization name string (UTF-16) for the certificate/signing request, no zero termination.

length 2 Byte	**tag = 6** 2 Byte
organization name (length - header) Bytes...	

16 32

Tag Unit Name

Contains the organizational unit name string (UTF-16) for the certificate/signing request, no zero termination. This tag is optional.

length 2 Byte	**tag = 7** 2 Byte
organizational unit name (length - header) Bytes...	

16 32

Tag Locality Name

Contains the locality name string (UTF-16) for the certificate/signing request, no zero termination. This tag is optional.

length 2 Byte	**tag = 8** 2 Byte
locality name (length - header) Bytes...	

16 32

Tag Country Name

Contains the country name string (UTF-16) for the certificate/signing request, no zero termination. This tag is optional.

length 2 Byte	**tag = 9** 2 Byte
country name (length - header) Bytes...	

16 32

Tag State or Province Name

Contains the state or province name string (UTF-16) for the certificate/signing request, no zero termination. This tag is optional.

length 2 Byte	**tag = 10** 2 Byte
state or province name (length - header) Bytes...	

16 32

Tag Password

Contains the password used for private key decryption, no zero termination. This tag is optional.

length 2 Byte	**tag = 11** 2 Byte
Password (length - header) Bytes...	

16 32

Tag Certificate Extension

With this tag you can provide one X.509 certificate extension which should be added to a self signed certificate or a certificate request (there the extension will be added as ExtensionRequest).
The extension must already be encoded in it's ASN.1 representation as defined for 'Extension' in RFC5280 chapter 4.1. This means a sequence containing an object identifier followed by a bitstring which contains the extension specific payload.
The ASN.1 structure is directly added to the certificate/certificate request, so the caller is responsible for proper content of the extension which will be provided by this tag.
If this tag is missing no extension will be added. It can occur a maximum of 5 times.

length 2 Byte	**tag = 12** 2 Byte
ASN.1 encoded extension	

(length - header) Bytes...
16 32

CONF_CERTIFICATE_REQUEST_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bed	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	4 bytes max label length, 4 bytes max IP string length, 4 bytes number of supported key types, 4 bytes number of supported cmd types, numKeyTypes * keyType (4 bytes type ID, 64 bytes type label), numCmdTypes * Type (4 bytes cmd type ID, 64 bytes cmd type label), 4 bytes max password length
Write	p_octet	access_right_minimal	not supported

CONF_CERTIFICATE_REQUEST_PROGRESS

Tag code	NumDes	Message	SNMP Support
0x0bf0	-	yes (Message only)	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	-	access_right_minimal	see detailed description

Payload Structure

length 2 Byte	**tag** 2 Byte
payload (length - header) Bytes	

16 32

Generic description

This message informs about the current status of certificate/signing request/key operations started by a CONF_CERTIFICATE_REQUEST.
The format has an internal tagged structure.

Tags

Status: tag = 0

Contains the command type and its status.

length 2 Byte	**tag = 0** 2 Byte
Type 4 Bytes	
Status 4 Bytes	

16 32

Type

Values:

create PKCS#10 CSR	0
create self-signed certificate	3
decrypt private key	4

Status

Values:

Values in the range of 0 to 100 represent the current status, other values need to be handled as error.
Status values:

Generating request	25
Sending request	50
Request completed successfully	100

Error values:

General error	129
File already exists error	130
No free key entry error	131
No storage space error	132
Server error	133
Decrypt error	134
Entry not found error	135
No certificate found for key error	136
Error parsing the key structure or key type not supported	137

Tags

Label: tag = 1

Contains the certificate/signing request/key label, no zero termination.

length	**tag = 1**

2 Byte	2 Byte
Label (length - header) Bytes...	

16 32

Tags

New label: tag = 2

Contains the certificate/signing request/key label, no zero termination.
This tag is optional and is used to transmit the new label that is used to store the entry after
the operation is complete (e.g. a key is decrypted and matched against a certificate and stored using the certificate label).

length 2 Byte	**tag = 1** 2 Byte
Label (length - header) Bytes...	

16 32

CONF_CERTIFICATE_USAGE

Tag code	NumDes	Message	SNMP Support
0x0bf2	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

length 2 Byte	**tag** 2 Byte
payload (length - header) Bytes	

16 32

Generic description

Command CONF_CERTIFICATE_USAGE allows to configure for which purpose the certificates and keys on the device are used.
The format has an internal tagged structure, the allowed tags and values can be queried via CONF_CERTIFICATE_USAGE_OPTIONS.
There are two types of certificate usages that can be distinguished by the highest bit of the certUsageID:
0: device certificate chains starting with a label identifying the device certificate and private key (both using the same label) followed by further chain certificates
1: lists of trusted certificates used to verify received peer certificates

Tags

certUsageID: tag = 0

Contains the certificate usage ID. This tag is mandatory for the read and write commands and should appear only once.

length 2 Byte	**tag = 0** 2 Byte
certUsageID 4 Bytes	

16 32

Tags

Certificate/key label: tag = 1

Contains the label string used to identify the certificate/key, no zero termination. Multiple label tags can be contained in the payload.

length 2 Byte	**tag = 1** 2 Byte
certificate/key label (length - header) Bytes...	

16 32

CONF_CERTIFICATE_USAGE_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bf3	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_minimal	not supported

Payload Structure

Write support 4 Bytes
Max label length 4 Bytes
Max number of labels 4 Bytes
Number of usages 4 Bytes
First usage entry (4 + "Max label length") Bytes

"Number of usages" x (4 + "Max label length") Bytes	**Usage entries N** (4 + "Max label length") Bytes ...

Write support

Certificate usage write support. Indicates wether it is possible to write the certificate usages.

Values:

write not supported	0
write supported	1

Max label length

Maximum length of one label string for the usages.

Values:

Max number of certificates

Defines the maximum number of certificates which can be associated with one usage.

Values:

Number of usages

Amount of usage entries following after this entry.

Values:

Usage entries 1 to "Number of usages"

Each entry consists of 4 byte usageID and the label of the length given in the "Max label length" field.

Values:

General Description

This command returns the list of certificate usages available in the camera. Each usage has a unique ID and a label which can be presneted to the end user. In the CONF_CERTIFICATE_USAGE command the usageID is referenced.

Flags in labelID

Some bits of the usage ID are reserved and contain additional information:

Values

Bit 31 (0x80000000)	0: Certificate is an end-user certificate. Certificates in these usages are presented to a peer for client authentication for example. These usages can only contain one certificate. The certificate must be associated with a private key stored on the device. 1: Certificate is a trusted certificate. Certificates in these usages are used to verify certificates presented from the external peer. These usage can contain multiple certificates. The certificate needs no private key on the device.
Bit 30 (0x40000000)	ATTENTION: This bit is reserved for future use. Planned for the following meaning: This bit is only valid, if bit 31 is 0. 0: This end-user certificate usage (bit 31 is 0) can only be associated with 1 certificate. 1: This end-user certificate usage (bit 31 is 0) can be associated with multiple certificate (not used or supported at the moment).

CONF_STREAM_SECURITY_V2

Tag code	NumDes	Message	SNMP Support
0x0bb8	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	reads rtp stream security settings (2 bytes hash id[0=off, 1=classic watermark, 2=MD5, 3=SHA1, 4=SHA256], 2 bytes signature id[0=Off, 1=RSAwithSHA1], 4 bytes signature interval)
Write	p_octet	access_right_service	configures rtp stream security settings (2 bytes hash id[0=off, 1=classic watermark, 2=MD5, 3=SHA1, 4=SHA256], 2 bytes signature id[0=Off, 1=RSAwithSHA1], 4 bytes signature interval, setting signature interval to 0xFFFFFFFF selects default)

CONF_STREAM_SECURITY_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bb9	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads rtp stream security settings options (2 bytes maxHashId, 2 bytes maxSigId, 4 bytes minSigDist, 4 bytes maxSigDist, 4 bytes minDistMid, 4 bytes minDistLow)
Write	p_octet	access_right_minimal	write not supported

CONF_VIDEO_OVER_SSL

Tag code	NumDes	Message	SNMP Support
0x0b65	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	returns if video over ssl is supported or not (0: disabled)
Write	t_dword	access_right_user	enables/disables video over ssl (0:disabled)

CONF_WIFI_CONFIG_TOKEN

Tag code	NumDes	Message	SNMP Support
0x0c29	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_internal	read WifiConfigToken; used for residential cloud services APP as a secret to change the Wifi network settings after initial configuration
Write	p_string	access_right_service	store WifiConfigToken; used for residential cloud services APP as a secret to change the Wifi network settings after initial configuration

CONF_LOGIN_LIMITER_MESSAGE

Tag code	NumDes	Message	SNMP Support
0x0cb3	no	yes	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_internal	neither read nor write access, only message, see detailed description
Write	p_octet	access_right_service	neither read nor write access, only message, see detailed description

Payload Structure

Number of Entries 2 Byte	**TotalLength** 2 Byte
Entry 1 x Bytes	
... y Bytes	
Entry N z Bytes	

Number of Entries

Number of Entries following

TotalLength

Total Length in Bytes, including all entries and this header

Entry

a tagged entry looks like this:

EntryID 2 Byte	**EntryLength N = 4 + n Bytes** 2 Byte
EntryPayload n Bytes	

EntryID

ID of the entry

ID	Tag	Payload Length	Payload Description
1	LOGIN_LIMITER_MSG_STATUS	1 byte	0: Fail Msg; 1: Success Msg
2	LOGIN_LIMITER_SERVER_NAME	variable n bytes	String with server name
3	LOGIN_LIMITER_REMOTE_ADDRESS	variable n bytes	String with remote address of access origin
4	LOGIN_LIMITER_USER_NAME	variable n bytes	String with user name of access

EntryLength

Length of the entry in bytes, including the payload, but also ID and length field

EntryPayload

Payload of the entry

CONF_NBR_OF_TEMP_SENS

Tag code	NumDes	Message	SNMP Support
0x09c4	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	the number of temperature sensors
Write	t_dword	access_right_minimal	not supported

CONF_TEMP_SENS

Tag code	NumDes	Message	SNMP Support
0x09c5	temperature sensor	yes	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_user	value of the temperature sensor specified by numdes
Write	t_dword	access_right_minimal	not supported

CONF_REDUNDANT_POWER

Tag code	NumDes	Message	SNMP Support
0x09d7	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	reads the redundant power settings 0=off, 1=on
Write	t_dword	access_right_service	sets the redundant power 0=off, 1=on

CONF_POWER_SUPPLY

Tag code	NumDes	Message	SNMP Support
0x09dd	no	yes	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_user	bit field of power supplies (supply n: supplybits = 1<<(n-1))
Write		access_right_service	Data type : t_dword

CONF_NBR_OF_FANS

Tag code	NumDes	Message	SNMP Support
0x09d5	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	reads the number of fans
Write	t_dword	access_right_service	not supported

CONF_FAN_SPEED

Tag code	NumDes	Message	SNMP Support
0x09d6	fan number	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_user	fan speed in RPM
Write	t_dword	access_right_service	not supported

CONF_MINIMUM_FAN_SPEED

Tag code	NumDes	Message	SNMP Support
0x09de	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	minimum fan speed in RPM
Write	t_dword	access_right_service	minimum fan speed in RPM

CONF_BOOT_STATE

Tag code	NumDes	Message	SNMP Support
0x09e3	no	no	no

	Datatype	Access Level	Description
Read	t_dword	unprotected	100 if booted, 50: waiting for camera frontend, 25: waiting for DHCP
Write	t_dword	access_right_service	not supported

CONF_UPLOAD_PROGRESS

Tag code	NumDes	Message	SNMP Support
0x0701	-	yes (Message only)	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	-	access_right_minimal	see detailed description

Payload Structure

Progress
1 Byte

8

Progress

informs about current progress of an upload or possible error states

Values

progress in percent	1 ... 100
header error	101
write error	102
read back error	103
verify error	104
checksum mismatch (read back flash content vs. written data)	105
checksum mismatch (announced in header vs. received data)	106
magic error	110
version too low	111
flash type incompatible	112
device check failed	113
file entry marker failed	114
chunk size error	115
area already written	116
black white list check	117
wrong or no signature	118
signature invalid	119
invalid file name	130
ROM init error	131
ROM write error	132
ROM close error	133
socket error	134
overflow error	135
final flush error	136
file format error	137
logo size error (size of picture is limited to 128x128 pixel)	138
logo compression error (only uncompressed bitmap files allowed)	139
logo colour error (max. 256 colour bitmap supported)	140
certificate/key already exists (filename must be unique)	141
certificate/key format error	142
no matching certificate found for uploaded key (upload certificate first or upload combined file)	143
no free key entry available	144
no certificate storage space available	145
device not fully booted	146
password required for this upload (encrypted config file)	147
no memory to store the file	148
file is to large for that type of upload	149

Values in the range of 1 ... 100 represent the upload progress in percent. Other values need to be handled as error during the upload and inform about the kind of the error that was encountered during the upload.

CONF_HEATER_MODE

Tag code	NumDes	Message	SNMP Support
0x0b0c	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	read the heater mode - 0: off, 1: on, 2: auto
Write	t_octet	access_right_service	write the heater mode - 0: off, 1: on, 2: auto

CONF_CHECK_DEFAULT_BUTTON_STATE

Tag code	NumDes	Message	SNMP Support
0x0c3b	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	check the current status of the (factory) default button. 0=inactive; 1=active
Write	%	access_right_service	%

CONF_CHECK_POS_FB_STATE

Tag code	NumDes	Message	SNMP Support
0x0c3c	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	check the current status of the position feedback sensors. (Only applicable on certain devices). bit 0: end position fb switch, bit 1: hall sensor
Write	%	access_right_service	%

CONF_CLOUD_COMMISSIONING_STATUS

Tag code	NumDes	Message	SNMP Support
0x0c44	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the status of cloud commissioning. After successful commissioning, the cloud sets this status so the camera knows about it (and can e.g. change LED accordingly). 0=uncommissioned; 100=commissioning successfully finished
Write	t_word	access_right_service	store the status of cloud commissioning. After successful commissioning, the cloud sets this status so the camera knows about it (and can e.g. change LED accordingly). 0=uncommissioned; 100=commissioning successfully finished

CONF_AMBIENT_LIGHT_LEVEL

Tag code	NumDes	Message	SNMP Support
0x0c45	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	get the normalized ambient light level (value range: 0-1000; typical values between 0-400); Only available on certain devices
Write	t_dword	access_right_service	not supported

CONF_LOW_AMBIENT_LIGHT_THRESHOLD

Tag code	NumDes	Message	SNMP Support
0x0c46	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	get the normalized low ambient light level threshold (value range: 0-1000; typical 50); Only available on certain devices; To obtain the current level, see CONF_AMBIENT_LIGHT_LEVEL
Write	t_dword	access_right_service	set the normalized low ambient light threshold (value range: 0-1000; typical 50); Only available on certain devices; To obtain the current level, see CONF_AMBIENT_LIGHT_LEVEL

CONF_NBR_OF_HUMIDITY_SENSORS

Tag code	NumDes	Message	SNMP Support
0x0c60	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	the number of humidity sensors
Write	t_dword	access_right_minimal	not supported

CONF_HUMIDITY_VALUE

Tag code	NumDes	Message	SNMP Support
0x0c61	humidity sensor (1...x)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	value of the humidity sensor specified by numdes (humidity 0 ... 100.00 % maps onto values 0 ... 10000)
Write	t_dword	access_right_minimal	not supported

CONF_VIPROC_ID

Tag code	NumDes	Message	SNMP Support
0x0803	video line	yes	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the viproc interface id (0:= no viproc)
Write	t_dword	access_right_service	not supported

CONF_RULE_ENGINE_ID

Tag code	NumDes	Message	SNMP Support
0x0b94	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	reads the version id of the rule engine
Write	t_dword	access_right_service	not supported

CONF_VIPROC_ONOFF

Tag code	NumDes	Message	SNMP Support
0x0800	video line	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	reads the VCA status 0=off, 1=on. command is obsolete with firmware 4.0 and replaced by VIPROC_MODE.
Write	flag	access_right_service	switches on/off the video content analysis (VCA). command is obsolete with firmware 4.0 and replaced by VIPROC_MODE.

CONF_VIPROC_CUSTOM_PARAMETERS

Tag code	NumDes	Message	SNMP Support
0x0802	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads the video content analysis (VCA) custom parameters, payload is beyond the scope of this document
Write	p_octet	access_right_service	sets the VCA custom parameters, payload is beyond the scope of this document

CONF_VIPROC_FIRE_PARAMETERS

Tag code	NumDes	Message	SNMP Support
0x0c0c	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	gets the VCA fire detector settings (no vca profiles), payload is beyond the scope of this document
Write	p_octet	access_right_service	sets the VCA fire detector settings (not part of the vca profiles), payload is beyond the scope of this document

CONF_VIPROC_CUSTOM_PARAMETERS_TAGS

Tag code	NumDes	Message	SNMP Support
0x0bac	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads one or more parameter tags from viproc configuration, payload is beyond the scope of this document
Write	p_octet	access_right_service	sets one or more tags in the VCA custom parameters, payload is beyond the scope of this document

CONF_VIPROC_GLOBAL_PARAMETERS

Tag code	NumDes	Message	SNMP Support
0x0b68	break;	ok = false;	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads the VCA global parameters, payload is beyond the scope of this document
Write	p_octet	access_right_service	sets the VCA global parameters, payload is beyond the scope of this document

CONF_VIPROC_GLOBAL_PARAMETERS_TAGS

Tag code	NumDes	Message	SNMP Support
0x0c68	break;	ok = false;	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads the VCA global parameters, payload is beyond the scope of this document
Write	p_octet	access_right_service	sets the VCA global parameters, payload is beyond the scope of this document

CONF_VIPROC_DLL_NAME

Tag code	NumDes	Message	SNMP Support
0x0804	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	reads the name of currently selected VCA component
Write	p_string	access_right_service	specify the name of the VCA component

CONF_VIPROC_DLL_NAME_LIST

Tag code	NumDes	Message	SNMP Support
0x0808	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	list of viproc component names
Write	p_string	access_right_service	list of viproc component names

CONF_VIPROC_SAVE_REFERENCE_IMAGE

Tag code	NumDes	Message	SNMP Support
0x0805	video line	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	not supported
Write	f_flag	access_right_service	triggers the VCA to create a reference image, but only if possible (e.g. not a dome camera)

CONF_VIPROC_REFERENCE_IMAGE_FILENAME

Tag code	NumDes	Message	SNMP Support
0x0806	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	reads the current reference image filename
Write	p_string	access_right_service	not supported

CONF_START_VIPROC_CONFIG_EDITING

Tag code	NumDes	Message	SNMP Support
0x0a38	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	Moves to the dome to the scene corresponding to the specified config and freezes the dome there. If already a config is in configuration mode, this command returns with a dword of 0. All subsequent viproc commands are related to this configuration. This command works also for non-dome devices. There it won't try to move the dome to the corresponding scene.

CONF_STOP_VIPROC_CONFIG_EDITING

Tag code	NumDes	Message	SNMP Support
0x0a39	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	Stops editing of the specified config (see CONF_START_VIPROC_CONFIG_EDITING). If the specified scene does not match the currently edited config, a dword with value 0 is returned.

CONF_CONT_VIPROC_CONFIG_EDITING

Tag code	NumDes	Message	SNMP Support
0x0a3a	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	Configuration of a config stops automatically after 20 seconds. This command can be used to keep configuration active for another 20 seconds. If the specified config is currently not in configuration mode, the command replys with a dword of 0.

CONF_VIPROC_SCENE

Tag code	NumDes	Message	SNMP Support
0x0a3b	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the scene for which the active config is defined.
Write	t_dword	access_right_service	writes the scene number of the active viproc configuration

CONF_NUMBER_OF_VIPROC_CONFIGS

Tag code	NumDes	Message	SNMP Support
0x0a3c	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of configurations for the specified line.
Write	t_dword	access_right_service	-

CONF_VIPROC_DEFAULT_LIST

Tag code	NumDes	Message	SNMP Support
0x0cba	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	List of available IVA default configuration sets, value: string| ... eg, 0=default|1=line_crsossing.iva|2=inside_field.iva
Write	p_string	access_right_service	write not suported

CONF_VIPROC_DEFAULT

Tag code	NumDes	Message	SNMP Support
0x0cbb	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read not supported
Write	t_octet	access_right_service	

Payload Structure

This commands deploys a dedicated config file to set VCA config. The list of available scenario defaults can be retrieved with the command VIPROC_DEFAULT_LIST. In case of an error this command may return with a command specific error code. For a list of the specific error code see below. The value associated to scenario will be kept constant.

Defaults Scenarios

0	default
1	Intrusion (one field)
2	Intrusion (two fields)
3	People counting
4	Traffic incidents
5	Traffic wrong way

Command specific error codes

0x01	Default scenario not available
0x02	Calibration missing
0xF0	Internal error

CONF_ACTIVE_VIPROC_CONFIG

Tag code	NumDes	Message	SNMP Support
0x0a3f	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the config id activated by bool engine.
Write	t_dword	access_right_service	Sets the specified config ID as active. The command has only an effect, if the VIPROC_MODE is set to SCRIPT mode (0xff). this command is not available on a gen4 dome.

CONF_LOADED_VIPROC_CONFIG

Tag code	NumDes	Message	SNMP Support
0x0a3d	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the currently loaded config id.
Write	t_dword	access_right_service	cannot be written.

CONF_VIPROC_CONFIG_NAME

Tag code	NumDes	Message	SNMP Support
0x0a40	video line	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	returns the name of the currently active viproc configuration.
Write	p_unicode	access_right_service	writes the name of the currently active viproc configuration. The name can have at most 15 unicode characters.

CONF_LIST_OF_VIPROC_SCENES

Tag code	NumDes	Message	SNMP Support
0x0a41	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	this command works only for Gen4 autodomes. It returns the list of defined scenes together with their names. Thereby the first byte specifies the scene number, followed by the length L of the scene name. The next L bytes belong to the scene name in unicode without 0 termination. This structure repeats until either the payload ends or the scene number is zero.
Write	t_dword	access_right_service	-

CONF_VIPROC_TAGGED_CONFIG

Tag code	NumDes	Message	SNMP Support
0x0a42	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read a list of viproc items in an atomar way. see detailed description .
Write	p_octet	access_right_service	writes any viproc commands in an atomar way. see detailed description .

This command is an extension of the already existing RCP+ commands CONF_VIPROC_DLL_NAME, CONF_VIPROC_ONOFF, CONF_VIPROC_SCENE, CONF_VIPROC_CONFIG_NAME, CONF_VIPROC_CUSTOM_PARAMETERS, CONF_VCD_OPERATOR_PARAMS, CONF_VIPROC_ALARM_MASK, CONF_VIPROC_ALARM_AGGREGATION_TIME, and CONF_VIPROC_MOTION_DEBOUNCE_TIME. Each of these commands can only access one specific item of the currently active viproc configuration. The new command allows an atomic access to an arbitrary subset of configuration items, not only of the currently active one but of any configuration of a line.

The payload has a tag structure which is as follows:

4	6	8	8+Length1				
Config ID 4 Bytes	**Tag1** 2 Bytes	**Length1** 2 Bytes	**Payload1** Length1 Bytes	**Tag2** 2 Bytes	**Length2** 2 Bytes	**Payload2** Length2 Bytes	**...**	**0** 4 Bytes
RCP Payload N Bytes								

where Config ID is an integer between 1 and the maximum number of configurations available for this line (this maximum number can be retrieved by the command CONF_NUMBER_OF_VIPROC_CONFIGS).

Num Value

The num value of the RCP command selects the video line to which the configuration belongs.

Return Value

For a read operation, payload of tags within the request are ignored. The reply payload contains all tags in the same order as in the request, however the size of the tags is adapted to fit the size of each item. If the structure of the request is invalid or an unknown tag is included, the command will fail.

For a write operation, the payload of tags must fit the size of the corresponding items. If this is not the case, the command will fail. If the CONF_VIPROC_CUSTOM_PARAMETERS and CONF_VCD_OPERATOR_PARAMS items are written for the currently active configuration, they are checked by the currently active viproc and ruleengine algorithms. If they are not accepted, the corresponding algorithm is set to default mode and the default parameters are put into the reply. To summarize, a write operation is successful, if the RCP+ command succeeds and the reply payload is identical to the request payload.

Tags

Values:

CONF_VIPROC_DLL_NAME
CONF_VIPROC_ONOFF
CONF_VIPROC_SCENE
CONF_VIPROC_CONFIG_NAME
CONF_VIPROC_CUSTOM_PARAMETERS
CONF_VCD_OPERATOR_PARAMS
CONF_VIPROC_ALARM_MASK
CONF_VIPROC_ALARM_AGGREGATION_TIME
CONF_VIPROC_MOTION_DEBOUNCE_TIME

Length

The length of the payload depends on the data type of the corresponding tag. The payload of a tag must be in network byte-order.

Values:

F_FLAG 1
T_WORD 2
T_DWORD 4
P_STRING string length
P_OCTET buffer length

The payload of a tag must be in network byte-order.

Example

The following payload example contains a configuration for the first configuration of a line. It consists of two tags, the first describing the CONF_VIPROC_CONFIG_NAME item, the second describing the CONF_VIPROC_ONOFF item.

4				6		8		15	17		19		20				24
Config ID 4 Bytes				**Tag1** 2 Bytes		**Length1** 2 Bytes		**Payload1** 7 Bytes	**Tag2** 2 Bytes		**Length2** 2 Bytes		**Payload2** 1 Byte	**0** 4 Bytes			
0x00	0x00	0x00	0x01	0x08	0x04	0x00	0x07	"viproc"	0x08	0x00	0x00	0x01	0x01	0x00	0x00	0x00	0x00
RCP Payload N Bytes																	

CONF_VIPROC_TAGGED_CONFIG_INTERNAL

Tag code	NumDes	Message	SNMP Support
0x0a43	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read a list of viproc items in an atomar way.
Write	p_octet	access_right_internal	writes any viproc commands in an atomar way.

CONF_VIPROC_ALARM

Tag code	NumDes	Message	SNMP Support
0x0807	video line	yes	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16	32	
Alarm Flags 2 Byte	**Detector Flags** 2 Byte	**ConfigId** 1 Byte
8	24	40

The message is sent whenever any of the bit values changes. Additionally it is sent once per 10 seconds when any of the bits is set.

Alarm Flags

Values:

0x8000	default task
0x4000	global change flag
0x2000	signal too bright flag
0x1000	signal too dark flag
0x0800	reserved
0x0400	image too blurry flag
0x0200	signal loss flag
0x0100	reference image check failed flag
0x0080	invalid configuration flag
0x0040	flame detection flag
0x0020	smoke detection flag

These bits can be combined by using the bitwise or-operator.

Detector

State of the detectors. 0x0001 corresponds to detector one, 0x0002 to detector two and so on.

ConfigId

Identifcation number of the VCA profile caused this alarm.

CONF_ALARM_OVERVIEW

Tag code	NumDes	Message	SNMP Support
0x0c38	video line (starts with 1)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16 32

Flags 1 Byte	**Reserved** 1 Byte	**Number of alarm entries** 2 Byte
Alarm Entry n Byte		
...		
Alarm Entry n Byte		

Messages are sent if a change has happened (event, state change, alarm added, alarm deleted). The readout command contains all registered alarm entries.

Flags

Values:

0x80 All entries included (read out)

Alarm Entry

16 32

Entry ID 2 Byte		**Entry Length** 2 Byte	
Flags 1 Byte	**Reserved** 1 Byte	**Alarm Source** 1 Byte	**Alarm Type** 1 Byte
Alarm Name n Byte			

8 24

Entry ID

Unique ID for each entry. May be reassigned to entry if entry is deleted.

Entry Length

Length of the entry payload, exclusive header.

Flags

Values:

0x80	Add Flag	Alarm Entry has been registered
0x40	Delete Flag	Alarm Entry has been unregistered
0x20	State Flag	Alarm Entry is a state
0x10	State Set Flag	Alarm Entry State is set (states only)

Alarm Source

Values:

0	unknown
2	Relais
3	Digital Input
4	Audio
5	Virtual Input
6	Tamper
7	Motion
8	Flow
9	Intelligent Video Analytics
10	Fire
11	Man Over Board

Alarm Type

Values:

0	unknown
1	VCA
2	Relais
3	Digital Input
4	Audio
5	Virtual Input
16	Default Task
17	Global Change
18	Signal too bright

19	Signal too dark
23	Reference Image Check Failed
24	Invalid Configuration
25	Flame Detected
26	Smoke Detected
32	Object in field
33	Crossing line
34	Loitering
35	Condition change
36	Following route
37	Tampering
38	Removed object
39	Idle object
40	Entering field
41	Leaving field
42	Similarity search
43	Crowd detection
44	Flow in field
45	Counter flow in field
46	Motion in field
47	Man over board
48	Counter
49	BEV People Counter
50	Occupacy

Alarm Name

String encoded in unicode.

CONF_AUPROC_ALARM

	Tag code	NumDes	Message	SNMP Support
	0x0a79	audio line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Message only,
Write	p_octet	access_right_service	-

Payload Structure

16	32
Alarm Flags 4 Byte	
8	24

Alarm Flags

The message is sent whenever any of the bit values changes. Additionally it is sent once per second when any of the bits is set.

Values:

Bit01	Frequency band 0-220 Hz
Bit02	Frequency band 110-360 Hz
Bit03	Frequency band 220-500 Hz
Bit04	Frequency band 360-690 Hz
Bit05	Frequency band 500-890 Hz
Bit06	Frequency band 690-1110 Hz
Bit07	Frequency band 890-1375 Hz
Bit08	Frequency band 1110-1690 Hz
Bit09	Frequency band 1375-2030 Hz
Bit10	Frequency band 1690-2420 Hz
Bit11	Frequency band 2030-2875 Hz
Bit12	Frequency band 2420-3400 Hz
Bit13	Frequency band 2875-4000 Hz
Bit14	Entire frequency band

Bit15-32 are unused and should be zero.

CONF_AUPROC_CONFIG

	Tag code	NumDes	Message	SNMP Support
	0x0a7a	yes (audio line)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	read configuration of AUPROC	
Write	p_octet	access_right_service	write configuration of AUPROC	

Payload Structure

16		32
Version 4 Byte		
Trigger Level 16 Byte		
Sensitivity 16 Byte		
unused 1 Byte	**Flags** 1 Byte	**reserved** 2 Byte
8		24

Description

This command configures the audio alarm analysis. The analysis is based on a 8kHz signal (if the device supports higher sampling rates the signal is downsampled).
The analysed frequency bands (there are 13 frequency bands) have the following center frequencies:
100 Hz, 220 Hz, 350 Hz, 500 Hz, 680 Hz, 880 Hz, 1.1 kHz, 1.4 kHz, 1.7 kHz, 2.0 kHz, 2.4 kHz, 2.9 kHz, 3.4 kHz
The freuqnecy bands are overlapping triangles.

Version

Unused (should be zero).

Trigger Level

16 Values between 0 and 255 (1 Byte each), setting the fixed threshold for every frequency band and energy level.
First Byte is for the lowest frequency and the last for the energy level. Trigger level 0 indicates that this frequency band is set off.
14 bytes (13 bands plus entire energie) are used in this version (refer to AUPROC_ALARM command).

Sensitivity

16 Values between 1 and 100 (1 Byte each), setting the sensitivity for every frequency band and energy level.
First Byte is for the lowest frequency and the last for the energy level.
14 bytes (13 bands pus entire energie) are used in this version.
The sensitivity is based on an adaptive threshold and a sliding window.

Flags

1 Byte for flags is reserved. The highest bit indicates audio alarm onoff.

Values:

Bit 8	OnOff flag

CONF_AUPROC_MELPEGEL

	Tag code	NumDes	Message	SNMP Support
	0x0a7b	yes (audio line)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the recent pegel in the frequency bands: detailed decription
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
recent melpegel 14 Byte	
adaptive threshold level 14 Byte	

Each of the 14 bytes corresponds to the frequency bands described within the command AUPROC_ALARM.

CONF_AUPROC_NAME

Tag code	NumDes	Message	SNMP Support
0x0a7c	audio line	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	-
Write	p_unicode	access_right_service	-

CONF_VIPROC_ALARM_MASK

Tag code	NumDes	Message	SNMP Support
0x0a23	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read which of the 32 bits produced by the viproc+ruleengine are considered for the CONF_VIPROC_ALARM. see CONF_VIPROC_ALARM for the meaning of the 32 bits.
Write	p_octet	access_right_service	configures which of 32 bits produced by the viproc+ruleengine are considered for the CONF_VIPROC_ALARM see CONF_VIPROC_ALARM for the meaning of the 32 bits.

CONF_VIPROC_CONFIG_CHANGE_IN_RECORDING

Tag code	NumDes	Message	SNMP Support
0x0a60	video line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	If the recording module detects a viproc configuration change, it sends this message. The payload of the message has the following structure: 8 Bytes - UTC timestamp int Time64 format, 1 Byte - config ID, 16 Bytes - MD5 Hash of the configuration.
Write	p_octet	access_right_service	-

CONF_VIPROC_MODE

Tag code	NumDes	Message	SNMP Support
0x0a65	video line	no	yes

	Datatype	Access Level	Description
Read	octet	access_right_minimal	switches between silent vca(0)/manual(1..16)/off(0xfd)/scheduler(0xfe)/script mode(0xff) of the video content analysis (VCA). In case of a moving dome and a preset with activated VCA profile the active Id is returned. (if vca is turned off then 0xfd is returned).
Write	octet	access_right_service	switches between silent vca(0)/manual(1..16)/off(0xfd)/scheduler(0xfe)/script mode(0xff) of the video content analysis (VCA). Command should not be used while configuring a profile (i.e. start_viproc_config_editing is active). On the gen4 dome, the modes 0xfe and 0xff are not supported. Any value between 0 and 16 is interpreted as on and 0xfd as off.

CONF_VIPROC_WEEKLY_SCHEDULE

Tag code	NumDes	Message	SNMP Support
0x0a66	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Returns the weekly VCA schedule of the specified line. The format of the payload is the same as the one of the CONF_HD_RECORD_SCHEDULE command.
Write	p_octet	access_right_service	Sets the weekly VCA schedule of the specified line. The format of the payload is the same as the one of the CONF_HD_RECORD_SCHEDULE command.

CONF_VIPROC_HOLIDAYS_SCHEDULE

Tag code	NumDes	Message	SNMP Support
0x0a67	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Returns the holidays VCA schedule of the specified line. The format of the payload is the same as the one of the CONF_HD_RECORD_HOLIDAYS command.
Write	p_octet	access_right_service	Sets the holidays VCA schedule of the specified line. The format of the payload is the same as the one of the CONF_HD_RECORD_HOLIDAYS command.

CONF_PTZ_CONTROLLER_AVAILABLE

Tag code	NumDes	Message	SNMP Support
0x0a51	video line	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	returns true if for the specified line a ptz controller is available which can be used by the VCA module.
Write	flag	access_right_service	not supported

CONF_ENABLE_VCA

Tag code	NumDes	Message	SNMP Support
0x0813	video line	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	This command is used to configure the dome. Note that we write the information in the second bit of ptz_controller_available.
Write	flag	access_right_service	not supported

CONF_VIPROC_ALARM_AGGREGATION_TIME

Tag code	NumDes	Message	SNMP Support
0x0aca	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	obsolete, please use VIPROC_ALARM_EXT_AGGREGATION_TIME.
Write	t_octet	access_right_service	obsolete, use VIPROC_ALARM_EXT_AGGREGATION_TIME.

CONF_VIPROC_ALARM_EXT_AGGREGATION_TIME

Tag code	NumDes	Message	SNMP Support
0x0ba7	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the alarm aggregation time of the video line (unit is 0.1 seconds, thus, 16 corresponds to 1.6 seconds). The return value has to bytes is a word beginning with the upper byte and followed by the lower byte.
Write	p_octet	access_right_service	write the duration of the alarm aggregation time in 0.1 seconds (1.6 sec coresponds to a value 16). The payload consists of an upper and a lower byte resulting in a maximal time of 6553.6 seconds. Note, this parameter belongs to a viproc config.

CONF_VIPROC_MOTION_DEBOUNCE_TIME

Tag code	NumDes	Message	SNMP Support
0x0acb	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read the motion alarm debounce time of the video line (4,3s == 43). Note, this parameter belongs to a viproc config.
Write	t_octet	access_right_service	write the duration of the motion alarm debounce time of the video line (1,6sec == 16). Note, this parameter belongs to a viproc config.

CONF_VCA_TASK_RUNNING_STATE

Tag code	NumDes	Message	SNMP Support
0x0a96	yes (camera line input number)	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	returs the current processing state of the VCA engine; TRUE when vca engine is running with no problems; FALSE if vca engine lack cpu performance
Write	-	access_right_service	not supported

CONF_VIPROC_VERSION

Tag code	NumDes	Message	SNMP Support
0x0ab2	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	viproc interface version on FPGA VCA devices
Write	p_octet	access_right_service	viproc interface version on FPGA VCA devices

CONF_TRANSPARENT_DATA_OVER_IP

Tag code	NumDes	Message	SNMP Support
0x0af2	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	set IP-address and port to accept transparent data, first, 4 bytes define the IP-address (0 represents off) and second, two bytes are the defined port
Write	p_octet	access_right_service	set IP-address and port to accept transparent data, first, 4 bytes define the IP-address (0 represents off) and second, two bytes are the defined port

CONF_AUTO_TRACKER_TRACK_OBJECT

Tag code	NumDes	Message	SNMP Support
0x0b2d	video line (starts with 1)	yes	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Get the object id of the currently tracked object. Zero is returned if no object is tracked.
Write	t_dword	access_right_user	send the object id to start tracking

CONF_AUTO_TRACKER_TRACK_OBJECT_POS

	Tag code	NumDes	Message	SNMP Support
	0x0c28	video line (starts with 1)	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	not supported	
Write	p_octet	access_right_user	send an image position to start tracking the nearest object. Position is in normalized coordinate system with upper left corner (0x0000 0x0000) and lower right corner (0x8000 0x8000)	

CONF_MODE_AUTO_TRACKER

	Tag code	NumDes	Message	SNMP Support
	0x0b40	video line (starts with 1)	yes	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	Get the autotracker mode (Modi: OFF=0, AUTO=1, CLICK=2).	
Write	t_octet	access_right_user	Sets the autotracker mode (Modi: OFF=0, AUTO=1, CLICK=2). The CLICK mode has to be rewritten after 10sec; otherwise, the mode shall switch back to AUTO	

CONF_STATUS_AUTO_TRACKER

	Tag code	NumDes	Message	SNMP Support
	0x0b86	-	yes	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	Gets current mode of auto tracker. 0=off; 1=idle; 2=seeking; 3=tracking active; 4=tracking passive	
Write	t_octet	access_right_user	Gets current mode of auto tracker.	

CONF_SENSITIVITY_OBJECT_BASED_VCA

Tag code	NumDes	Message	SNMP Support
0x0b31	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	reads the sensitivity of the object based VCA algorithm (of active config profile), if not an object based VCA algorithm is running the command returns with an error
Write	t_octet	access_right_service	sets the sensitivity of the object based VCA algorithm in the active config profile, if not an object based VCA algorithm is running the command returns with an error

CONF_VIPROC_SENSITIVE_AREA

Tag code	NumDes	Message	SNMP Support
0x0b78	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	sets the sensitive area of the object based VCA algorithm in the active (or last loaded) config profile, see detailed description
Write	p_octet	access_right_service	sets the sensitive area of the object based VCA algorithm in the active (or last loaded) config profile

Payload Structure

16 32

Width 2 Byte		**Height** 2 Byte	
Cells X 1 Byte	**Cells Y** 1 Byte	**Step X** 1 Byte	**Step Y** 1 Byte
Start X 1 Byte	**Start Y** 1 Byte	**Bitmask** n Byte	
Bitmask ...			

8 24

This command reads/sets the sensitivity mask of the viproc algorithm. Please use the read functionality to get the dimensions. If no viproc algorithm is running the command fails.

Width

Image width of the viproc algorithm.

Height

Image height of the viproc algorithm.

Cells X/Y

Number of cells in X/Y direction.

Step X/Y

Number of cell size in X/Y direction.

Start X/Y

Offset of bit mask from upper left corner.

Bitmask

Bitstream which signals which cell is activated or not. Upper left cell is the first going to the left and then down.
Example: A 3x3 mask with cross activated
oxo
xxx
oxo
has the following binary 0x010111010b or hexadecimal 0xba00h structure. Alignment bits can be set to zero but will be ignored.

CONF_REFERENCE_IMAGE_CHECK_INFO_MESSAGE

Tag code	NumDes	Message	SNMP Support
0x0b42	video line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	1st Byte (Bit 7: check failed, Bit 6: No info available (reference image check was not enabled or reference image has not enough contours to perform reference image check), Bits 5-0: reserved), 2nd Byte: correlation, 3rd Byte: correlation threshold, 4th-5th Byte: seconds to alarm;
Write	p_octet	access_right_service	not supported

CONF_IVA_COUNTER_VALUES

Tag code	NumDes	Message	SNMP Support
0x0b4a	video line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	reset all counters

This command, which is at the same time a message, is used to retrieve the list of configured counters with the their current values. The output has the following structure: Initially the number of configured counters is set. Afterwards for each counter the data consisting of counter id, counter type, counter name in unicode and counter value is provided.

Request payload structure

Number of Counters 1 Byte
Data of Counter i 70 Byte
Data of Counter i + 1 70 Byte
... 70 Byte

Number of Counters

This values gives the number of counters available in the payload. In the subsequent the data of each counter are sequentially provided.

Data of Counter

16

Id	Type	Name	Value
1 Byte	1 Byte	64 Byte	4 Byte

8 528

Id

Each counter has a unique id.

Type

Not supported yet.

Name

A unique name can be assigned to a counter, which is stored in unicode (maximum: 32 characters).

Value

The current counter value (DWORD).

CONF_BEST_FACE

Tag code	NumDes	Message	SNMP Support
0x0b6e	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	

Payload Structure

16			32
enable 1 Byte	**reserved** 1 Byte	**accountId** 1 Byte	**pictureFormat** 1 Byte
targetObjectWidth 4 Bytes			
time out 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
reserved 8 Bytes ...			
8		24	

enable

0: best face is disabled 1: best face is enabled

reserved

Must be written as zero

accountId

Select account which is used as target for the detected faces for detailed information's see CONF_ACCOUNT_SETTINGS

pictureFormat

Select file format 0: Jpeg 1: YUV420 2: uncompressed tiff

targetObjectWidth

Configure maximum face width in pixel 0 means auto no scaling is necessary delivers best performance

time out

Time out in seconds. If the time out is set to 0 the best face of the tracking process will be posted when the face has left the scene.
Defines a timeout for the maximum delay which is introduced till a face is posted. After a face is posted the tracking is restarted.
Example:
Face is in scene for 50 sec t = [0s,50s]
timeout is configured to 10 s
best face of [0s,10s] is posted
best face of [10s,20s] is posted
best face of [20s,30s] is posted
best face of [30s,40s] is posted
best face of [40s,50s] is posted

reserved

Must be written as zero

CONF_BEST_FACE

Allows the upload of detected faces to an ftp server or drop box account.

CONF_WRITE_XML_TO_VCD

Tag code	NumDes	Message	SNMP Support
0x0baf	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	not supported
Write	p_string	access_right_service	write payload to vcd stream within a XML VCD-tag, data can be refound in VCD layer 0 and VCD layer 14. The reply is a UTC time (first 4 bytes are the seconds since 2000 followed by two bytes residual milliseconds). The payload is limitted to 1400 bytes. A very simple xml validation is performed -> tags can only have 31 bytes length and nesting is limited, CDATA is not allowed. Everything before the first xml element and after the last xml element is skipped. Only utf 8 encoding is supported.

CONF_SCHEDULED_PTZ_PROFILES

Tag code	NumDes	Message	SNMP Support
0x0c1f	cam 1 ... n	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read scheduled ptz profiles of one cam, see detailed description
Write	p_octet	access_right_service	write scheduled ptz profiles of one cam, see detailed description

Payload Structure

PTZ Profile 1 8 Bytes ...

8 x 8 Bytes	**PTZ Profile 10** 8 Bytes ...

PTZ Profile 1 - 10

Upto 10 Scheduled PTZ profiles (at least 1) for the camera specified by the num parameter, see detailed description for a scheduled PTZ profile

SCHEDULED_PTZ_PROFILES

Read/Write upto ten scheduled ptz profiles (at least one) for a camera, payload upto 80 bytes total. The profiles will be written to the config. Based on the record schedule the profiles will become active and a PTZ/ROI Tour/Preset will be activated or deativated. The flag SCHEDULED_PTZ_PROFILE_FLAG_TOUR, will use a PTZ tour on PTZ domes. If the cam isn't a PTZ camera, ROI is used instead, if the line is capable of it. The preset/tour will be applied on changes only, except the SCHEDULED_PTZ_PROFILE_FLAG_RETRIGGER flag is set, this will repeat the apply periodically. An User will be always able to change the PTZ state on a camera after the scheduled apply.

scheduled PTZ profile structure

16 32

PTZ/ROI Preset/Tour 2 Byte	**flags** 1 Byte	**reserved** 1 Byte
reserved 4 Bytes		

8 24

PTZ/ROI Preset/Tour

number of a preset or tour for PTZ or ROI depending on the flags settings, if the tourflag is set, following values are defined, 1 - Tour A, 2 - Tour B, 3 - Custom Tour

flags

Flags:

SCHEDULED_PTZ_PROFILE_FLAG_TOUR	0x01	if set, the preset number specifies a PTZ tour (only for PTZ domes)
SCHEDULED_PTZ_PROFILE_FLAG_RETRIGGER_LOCK	0x02	if set, the preset (not tour) setting will be reapplied about every 10th second and ptz from user will be locked for both tour and preset (except bicom aux)

CONF_SCHEDULED_PTZ_PROFILE

Tag code	NumDes	Message	SNMP Support
0x0c20	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read one scheduled ptz profile, see detailed description
Write	p_octet	access_right_service	write one scheduled ptz profile, see detailed description

Payload Structure

Command Header 4 Bytes
PTZ Profile 8 Bytes
...

Command Header

Command header specifies the requested PTZ Profile, see detailed description for command header

PTZ Profile

One PTZ profiles for the camera specified by the command header, see detailed description for a scheduled PTZ profile

SCHEDULED_PTZ_PROFILE

Read/Write one scheduled ptz profiles for a camera, payload upto 12 bytes total. The profiles will be written to the config. Based on the record schedule the profiles will become active and a PTZ/ROI Tour/Preset will be activated or deativated. The flag SCHEDULED_PTZ_PROFILE_FLAG_TOUR, will use a PTZ tour on PTZ domes. If the cam isn't a PTZ camera, ROI is used instead, if the line is capable of it. The preset/tour will be applied on changes only, except the SCHEDULED_PTZ_PROFILE_FLAG_RETRIGGER flag is set, this will repeat the apply periodically. An User will be always able to change the PTZ state on a camera after the scheduled apply. In read direction only the command header (4 bytes) is required. The response will have the full payload size of 12 bytes including the PTZ profile.

Command Header

16 32

Camera 2 Byte	**reserved** 1 Byte	**Profile** 1 Byte

8 24

Camera

Camara line of the requested PTZ profile (1 to n)

Profile

Profile number of the requested PTZ profile (1 to 10)

scheduled PTZ profile structure

16 32

PTZ/ROI Preset/Tour 2 Byte	**flags** 1 Byte	**reserved** 1 Byte
reserved 4 Bytes		

8 24

PTZ/ROI Preset/Tour

number of a preset or tour for PTZ or ROI depending on the flags settings, if the tourflag is set, following values are defined, 1 - Tour A, 2 - Tour B, 3 - Custom Tour

flags

Flags:

SCHEDULED_PTZ_PROFILE_FLAG_TOUR	0x01	if set, the preset number specifies a PTZ tour (only for PTZ domes)
SCHEDULED_PTZ_PROFILE_FLAG_RETRIGGER_LOCK	0x02	if set, the preset (not tour) setting will be reapplied about every 10th second and ptz from user will be locked for both tour and preset (except bicom aux)

CONF_FIRE_ALARM

Tag code	NumDes	Message	SNMP Support
0x0c58	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	not supported

Payload structure

Flags
4 Byte

32

Read the current state of each single fire alarm.

Alarm Flags

Values (Byte 1):

0x01	flame alarm
0x02	smoke alarm
0x04	fire trouble: illumination out of range
0x08	general fire trouble

These bits can be combined by using the bitwise or-operator.

CONF_VCA_MSK_POLY

Tag code	NumDes	Message	SNMP Support
0x0c6e	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	configure polygon VCA mask (aka 'virtual mask')

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_VCA_MSK_OPTIONS.
Every tagged command consists of an id and a length field + payload.

Tags

Polygon description id: 2
use CONF_VCA_MSK_OPTIONS to query if supported (Byte[7] rcp vca mask support @ dome)

Configure up to n polygons query n via CONF_VCA_MSK_OPTIONS.
CONF_VCA_MSK_OPTIONS also defines if only rectangles are supported then only rectangles are allow to be send with this command.
Only one polygon is allowed to be send at once. (One poly relates in the dome case to one position configured via CONF_VCA_MASK_DOME_PTZ_POS)
Currently only non self intersecting polygons are supported refer to CONF_VCA_MSK_OPTIONS.

length = nrPoints*4+12 2 Byte		**tag = 2** 2 Byte	
enable 1 Byte	**id** 1 Byte	**nrPoints** 1 Byte	**reserved** 1 Byte
reserved 4 Bytes			
vertex[0].x 2 Byte		**vertex[0].y** 2 Byte	
vertex[1].x 2 Byte		**vertex[1].y** 2 Byte	
... 2 Byte		**...** 2 Byte	
vertex[nrPoints-1].x 2 Byte		**vertex[nrPoints-1].y** 2 Byte	

enable: 0: vca mask is not active 1: vca mask is active
id: 0... max number vca masks (defined via CONF_VCA_MSK_OPTIONS)
nrPoints: 0... max number points (defined via CONF_VCA_MSK_OPTIONS)
reserved must be written as 0

vertex[i].x: 0...32768
vertex[i].y: 0...32768

Definition of the coordinate system for vca masking (stick to sei information)
Upper left edge : (0,0)
Upper right edge: (32768,0)
Lower left edge : (0,32768)
Lower right edge: (32768,32768)

Tags

global vca mask options at dome id: 3
use CONF_VCA_MSK_OPTIONS to query if supported (Byte[7] rcp vca mask support @ dome)

length = 12 2 Byte		**tag = 3** 2 Byte	
disableMasks 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
reserved 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

8 16 24 32

reserved must be written as 0
disableMasks: disable all vca masks

CONF_VCA_MSK_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0c6f	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] reserved; Byte[1] rectangleSupport [0 no, 1 yes]; Byte[2] polygonSupport [0 no, 1 yes but non self intersecting]; Byte[3] maxNrOfVertices maximum points for polygon definition; Byte[4] maxNrOfVcaMasks per Line; Byte[5]-Byte[6] reserved; Byte[7] rcp vca mask support @ dome; Byte[8] reserved; Byte[9] nrVcaMskLines; Byte[10]-Byte[15] reserved
Write	p_octet	access_right_service	write not supported

CONF_VCA_MASK_DOME_PTZ_POS

Tag code	NumDes	Message	SNMP Support
0x0c70	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Only for internal use
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_VCA_MSK_OPTIONS.
Every tagged command consits of an id and a length field + payload.

Tags

vca mask dome ptz cmd: 0

length = 12 2 Byte		**tag = 0** 2 Byte	
command 1 Byte	**maskId** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
reserved 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

8 16 24 32

command:
moveDomeToDefinedPos = 0 move dome to position stored at maskId
storeCurrentDomePos = 1 associate id defined in maskId to current dome position
maskId
use CONF_VCA_MSK_OPTIONS to query number of supported masks; @CONF_VCA_MSK_OPTIONS Byte[4] maxNrOfVcaMasks per Line
define maskID which is used in conjunction with the command

CONF_SYSLOG_LOG_LEVEL

Tag code	NumDes	Message	SNMP Support
0x0c5e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get the SYSLOG log level EMERGENCY=0 .. DEBUG=7 RFC 5424; value must be one above the highest allowed event, so 0= all off
Write	t_octet	access_right_service	set the SYSLOG log level EMERGENCY=0 .. DEBUG=7 RFC 5424; value must be one above the highest allowed event, so 0= all off

CONF_SYSLOG_PROTOCOL

Tag code	NumDes	Message	SNMP Support
0x0c5f	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get the SYSLOG server protocol (0 = UDP, 1=TCP, 2=TLS)
Write	t_octet	access_right_service	set the SYSLOG server protocol (0 = UDP, 1=TCP, 2=TLS)

CONF_SYSLOG_HOST_STR

Tag code	NumDes	Message	SNMP Support
0x0950	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the SYSLOG host ip address
Write	p_string	access_right_service	set the SYSLOG host ip address

CONF_SYSLOG_PORT

Tag code	NumDes	Message	SNMP Support
0x0c69	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	reads the SYSLOG host network port
Write	t_word	access_right_service	set the SYSLOG host network port

CONF_SYSLOG_MESSAGE

Tag code	NumDes	Message	SNMP Support
0x0caa	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	not supported
Write	p_string	access_right_service	Write a message string to the SYSLOG system

CONF_CONFIG_SEALING_ENABLED

Tag code	NumDes	Message	SNMP Support
0x0c8a	Message : no	Description : get enabled state of config sealing 0=sealing disabled; 1=sealing enabled; 2=seal broken	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	t_octet	access_right_service	enable config sealing 0=sealing disabled; 1=sealing enabled; 2=seal broken (2 for internal write)

CONF_CONFIG_SEALING_STATUS

Tag code	NumDes	Message	SNMP Support
0x0c8c	no	yes	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	return status of config seal,
Write	p_octet	access_right_service	read only

Payload Structure

16	32
Status 1 Byte	**Reserved** 3 Bytes
SealSetTimestamp 4 Bytes	
SealRandom 4 Bytes	
SystemTimestamp 4 Bytes	

8 24

Status

Values:

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

Reserved

Reserved, should be ignored

Values:

SealSetTimestamp

Timestamp when seal was activated in seconds since year 2000

SealRandom

Random number generated once when seal was activated

Values:

SystemTimestamp

Current system time in seconds since year 2000

Values:

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 hole system configuration is completed CONFIG_SEALING_ENABLED need to be set to 1 (= Seal enabled).
In this state each critical configuration change causes the device to send a CONFIG_SEALING_STATUS message.

If a client wants to active verify if a seal is valid then it should read CONFIG_SEALING_STATUS
Then it should check its content for:
1. Status = 1 (Sealing valid)
2. SealSetTimestamp if the time contained here lies before the date of the last known installation change.
3. SystemTimestamp if it is about the actual local time (to protect against time manipulation on the SealSetTimestamp)

CONF_CLOUD_WATCH_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x0cd6	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns cloud watch settings of the device in a tagged format
Write	p_octet	access_right_service	

lists account tags.

Tagged Payload Structure (T(ag) L(ength) V(alue)

16	32
num Entries 4 Bytes	
tag1 2 Byte	**length1** 2 Byte
payload1 length1 bytes	
...	
tagN 2 Byte	**lengthN** 2 Byte
payloadN lengthN bytes	

num Entries
The total number of tagged entries.

length
The length of the payload. The payload of a tag must be in network byte-order.

tags
The following tags are defined:

Tag Name	Tag ID	Tag Type	Tag description
ACCOUNT_TAG_TYPE	1	UInt32	0: not configured, 1: ftp, 2: dropbox, 5: amazon s3, 6: kinesis
ACCOUNT_TAG_NAME	2	PChar	account name
ACCOUNT_TAG_URL	3	PChar	url, e.g url to the ftp server
ACCOUNT_TAG_LOGIN	4	PChar	username
ACCOUNT_TAG_PASSWORD	5	PChar	password
ACCOUNT_TAG_PATH	6	PChar	path
ACCOUNT_TAG_FLAGS	7	UInt32	optional flags (e.g. FTP Encryption mode 0=off,1=TLS)
ACCOUNT_TAG_ACCESS_KEY	8	PChar	aws access key (max 128 bytes)
ACCOUNT_TAG_SECRET_KEY	9	PChar	aws secret access key (max 128 bytes)
ACCOUNT_TAG_BUCKET_NAME	10	PChar	bucket name (amazon s3), max 128 bytes
ACCOUNT_TAG_REGION	11	PChar	aws region
ACCOUNT_TAG_CAMERA_ID	12	PChar	s3 camera id, string will be added to the uploaded filename (max 128 bytes)
ACCOUNT_TAG_FILE_DURATION	13	PChar	s3 file duration in seconds
ACCOUNT_TAG_STREAM_NAME	14	PChar	stream name
ACCOUNT_TAG_KBPS	15	UInt32	maximum datarate for the backup, see CONF_BACKUP_MAX_KBPS
ACCOUNT_TAG_GROUP	16	PChar	cloud watch log group
ACCOUNT_TAG_CLOUD_WATCH_ENABLED	17	UInt8	cloud watch enabled/disabled

CONF_SYSUPTIME

Tag code	NumDes	Message	SNMP Support
0x00b9	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the system uptime in seconds
Write	t_dword	access_right_service	read-only

CONF_SD_CARD_STATUS

Tag code	NumDes	Message	SNMP Support
0x0b89	SD card slot ID. Starts with 1.	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	Read the current status of the SD card. BIT 0: 0: no card, 1: card detected. BIT 1: 0: SD, 1: CF. BIT 2..15: reserved.
Write	void	access_right_service	not supported

CONF_OEM_DEVICE_NAME

Tag code	NumDes	Message	SNMP Support
0x097c	no	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the oem device name
Write	p_string	access_right_service	not supported

CONF_OEM_EXT_ID

Tag code	NumDes	Message	SNMP Support
0x097d	no	no	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	DEPRECATED: please use CONF_DEVICE_TYPE_IDS instead. get the oem extension id
Write	t_octet	access_right_service	not supported

CONF_OEM_DEVICE_DOMAIN

Tag code	NumDes	Message	SNMP Support
0x09e9	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the oem device domain
Write	p_string	access_right_service	get the oem device domain

CONF_CAMERA_POSITION

Tag code	NumDes	Message	SNMP Support
0x0bdf	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_CAMERA_POSITION_OPTIONS
(Byte[0] support local cartesian coordinate system [0 no, 1 yes];) 1 -> tag 1 and 2 are suported
(Byte[1] support wgs 1984 coordinate system [0 no, 1 yes]) 1 -> tag 3 and 4 are suported

Only a single set of coordinates can be used e.g. local coordinates or global WGS coordinates
it is possible to define the coordinates via a string or a binary command
Tag 1 (binary) and 3 (string) configure a local cartesian coordinate system.
Tag 2 (binary) and 4 (string) configure a wgs 1984 coordinate system.

Read will always return both variant of one configuration (string + binary representation)

Tags

Camera position undefined: 0

Default value when no camera position is configured can also be used to remove the camera position configuration

length = 4 2 Byte	**tag = 0** 2 Byte

16 32

Tags

Local cartesian camera position (binary): 1

length = 16 2 Byte	**tag = 1** 2 Byte
X 4 Bytes	
Y 4 Bytes	
Z 4 Bytes	

16 32

X, Y, Z s(32): specify the position of the camera in a local cartesian coordinate system in units of 1/(2^16) m

Tags

Local cartesian camera position (string): 3

length = 4+n 2 Byte	**tag = 3** 2 Byte
X;Y;Z n Bytes	

16 32

X, Y, Z (string: 5.4;5.4;5.4)): specify the position of the camera in a local cartesian coordinate system max: 32767 m min: -32767 m
Example:
XYZ = "1000.124;145.3434;123.2355"
-> x= 1000.124 m, y = 145.3434 m, z = 123.2355

Tags

WGS 1984 camera position (binary): 2

Define the camera position based on wgs 1984 (World Geodetic System 1984)

length = 24 2 Byte	**tag = 2** 2 Byte
longitude [64:32] 4 Bytes	
longitude [31:0] 4 Bytes	
latitude [64:32] 4 Bytes	
latitude [31:0]	

4 Bytes **height** 4 Bytes

16 32

longitude u(64): Specifies the longitude in units of (2*pi/(2^64))
latitude u(64): Specifies the latitude in units of (2*pi/(2^64))
height s(32): Signed value height above sea level in units of 1/(2^16) m

Tags

WGS 1984 camera position (string): 4

Define the camera position based on wgs 1984 (World Geodetic System 1984)

length = 4+n 2 Byte	**tag = 4** 2 Byte
longitude;latitude;height n Bytes	

16 32

longitude,latitude,height string(3.12;3.12;5.4): Specifies the longitude, latitude in degree allowed values are [-360.0...0.0...360]
also the height above sea level is defined limits are (max: 32767 m min: -32767)
Example: longitude;latitude;height="12.45674964;45.456464;125.1234"
-> longitude = 12.45674964°, latitude = 45.456464°, 125.1234 m

CONF_CAMERA_ORIENTATION

Tag code	NumDes	Message	SNMP Support
0x0be5	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	tag 2 Byte
payload length Bytes -4	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_CAMERA_POSITION_OPTIONS.(support orientation calibration[0 no support, 1 pan only, 2 full support])
1: pan only: Tag 1 and 3 are supported
2: full support: Tag 1,2,3 and 4 are supported

All commands can be send in binary or string format. Only one configuration is valid.
Tag 1 (binary) and 3 (string) configure only pan angle for configuration tag 1 or 3 can used.
Tag 2 (binary) and 4 (string) configure can be used to configure the full camera orientation.

Read will always return both variant of one configuration (string + binary representation)

Tags

Camera orientation undefined: 0

Default value when no camera orientation is configured, can also be used to remove the camera orientation configuration

length = 4 2 Byte	tag = 0 2 Byte

16 32

Tags

Configure camera orientation only pan angle can be configured (legacy mode: Plane is configured via VCA config): 1

length = 8 2 Byte	tag = 1 2 Byte
pan 4 Bytes	

16 32

pan u(32): Specifies the pan angle in units of (2*pi/(2^32))

Tags

Camera orientation configuration via string, only pan angle can be configured (legacy mode: Plane is configured via VCA config): 3

length = 4+n 2 Byte	tag = 3 2 Byte
pan n Bytes	

16 32

pan string (string 3.6): Specifies the pan angle (in degrees) as a string allowed values are [-360.0...0.0...360.0]
Example string:
pan="3.455"
-> pan = 3.455°

Tags

Full camera orientation configuration: 2

length = 16 2 Byte	tag = 2 2 Byte
pan 4 Bytes	
tilt 4 Bytes	
roll 4 Bytes	

16 32

pan u(32): Specifies the pan angle in units of (2*pi/(2^32))
tilt u(32): Specifies the tilt angle in units of (2*pi/(2^32))
roll u(32): Specifies the roll angle in units of (2*pi/(2^32))

Tags

Full camera orientation configuration (via string): 4

length = 4+n 2 Byte	**tag = 4** 2 Byte
pan;tilt;roll n Bytes	

16 32

pan string (string: 3.6;3.6;3.6): Specifies the pan;tilt;roll angle (in degrees) as a string allowed values are [-360.0...0.0...360.0]
Example string:
pan;tilt;roll="1.1456;145.455;14.01"
-> pan = 1.1456°, tilt = 145.455°, roll = 14.01°

CONF_CAMERA_POSITION_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0be0	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] support local cartesian coordinate system [0 no, 1 yes]; Byte[1] support wgs 1984 coordinate system [0 no, 1 yes]; Byte[2] support orientation calibration[0 no support, 1 pan only, 2 full support]; Byte[3-7]reserved
Write	p_octet	access_right_service	not supported

CONF_CAMERA_SURROUNDING

Tag code	NumDes	Message	SNMP Support
0x0be1	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	tag 2 Byte
payload length Bytes -4	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_CAMERA_SURROUNDING_OPTIONS
([0: no, 1: flat plane only elevation can be configured (plane is assumed to be parallel to the (world, local) coordinate system])
1 -> Tag 1 (binary) and 3 (string) configure a offset to the ground plane (orientation is derived from (CONF_CAMERA_ORIENTATION))

Only a single surrounding configuration
it is possible to define the coordinates via a string or a binary command
Tag 1 (binary) and 3 (string) configure a elevation to the ground plane under the assumption that the ground plane
is parallel to the camera surrounding defined by (CONF_CAMERA_POSITION and CONF_CAMERA_ORIENTATION)
surrounding is defined by CONF_CAMERA_ORIENTATION [tilt, roll] + CONF_CAMERA_SURROUNDING [elevation]
Read will always return both variant of one configuration (string + binary representation)

Tags

Camera surrounding undefined: 0

Default value when no camera surrounding is configured, can also be used to remove the camera surrounding configuration

length = 4 2 Byte	tag = 0 2 Byte

16 32

Tags

Camera ground model plane is parallel to coordinate system defined by (CONF_CAMERA_POSITION and CONF_CAMERA_ORIENTATION)(binary): 1

length = 8 2 Byte	tag = 1 2 Byte
elevation 4 Bytes	

16 32

elevation s(32): specify the elevation relativ to the ground plane in units of 1/(2^16) m

Tags

Camera ground model plane is parallel to coordinate system defined by (CONF_CAMERA_POSITION and CONF_CAMERA_ORIENTATION) (string): 3

length = 4+n 2 Byte	tag = 3 2 Byte
elevation n Bytes	

16 32

elevation(string: 5.4)): specify the position of the camera in a local cartesian coordinate system max: 32767 m min: -32767 m
Example:
elevation = "1000.124"
-> elevation= 1000.124 m

CONF_CAMERA_SURROUNDING_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0be2	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] support plane as surrounding model[0: no, 1: flat plane only elevation can be configured (plane is assumed to be parallel to the (world, local) coordinate system]; Byte[1-7]reserved
Write	p_octet	access_right_service	not supported

CONF_CAMERA_LOCATION_METADATA

Tag code	NumDes	Message	SNMP Support
0x0bfa	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4	

16 32

Generic description

Command CONF_CAMERA_LOCATION_METADATA allows to configure/retrieve several descriptive location metadata fields, e.g. City, Country Code. The command has an internal tagged structure.

Tags

City: tag = 0

Contains the city string, no zero termination.

length 2 Byte	**tag = 0** 2 Byte
city string (length - header) Bytes...	

16 32

Tags

Zip code: tag = 1

Contains the zip code string, no zero termination.

length 2 Byte	**tag = 1** 2 Byte
zip code string (length - header) Bytes...	

16 32

Tags

Country code: tag = 2

Contains the country code string, no zero termination.

length 2 Byte	**tag = 2** 2 Byte
country code string (length - header) Bytes...	

16 32

Tags

Region: tag = 3

Specifies the region.

length 2 Byte	**tag = 3** 2 Byte
region 1 Byte	

16 32

CONF_CAMERA_CALIBRATION

Tag code	NumDes	Message	SNMP Support
0x0c34	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	calculates calibration with calibration elements, payload is beyond the scope of this document
Write	p_octet	access_right_service	calculates calibration with calibration elements, payload is beyond the scope of this document

CONF_COMPLETE_CALIBRATION_ELEMENT

Tag code	NumDes	Message	SNMP Support
0x0c35	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	4 bytes reserved, calibration element
Write	p_octet	access_right_service	write not supported

CONF_SHIFT_CALIBRATION_ELEMENT

Tag code	NumDes	Message	SNMP Support
0x0c82	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	4 bytes reserved, 2 bytes shiftx (0x8000 normalized), 2 bytes shifty (0x8000 normalized), calibration element
Write	p_octet	access_right_service	write not supported

CONF_SENSOR_ORIENTATION

Tag code	NumDes	Message	SNMP Support
0x0c39	IMU Sensor at image sensor: 0x0001-0x00fe; IMU Sensor at chassis: 0x0101 ... 0x01ff; legacy mapping: 0 -> 1, 0xff -> 0x0101, 0x00ff -> 0x0101	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	
Write	p_octet	access_right_service	not supported

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4	

16 32

Tagged format

The format has an internal tagged structure with the following tags:
1: pan
2: tilt
3: roll
4: rotate
5: vector

Missing tags are to be considered as unknown (e.g., no sensor available)

PanTag

Pan angle (magnetic sensor present).

length = 12 2 Byte	**tag = 1** 2 Byte
pan 4 Bytes	
error_est 4 Bytes	

16 32

pan u(32): Specifies the pan angle in units of (2*pi/(2^32))
error_est u(32): Specifies the estimation error in units of (2*pi/(2^32))

TiltTag

Tilt angle (acceleration sensor present).

length = 12 2 Byte	**tag = 2** 2 Byte
tilt 4 Bytes	
error_est 4 Bytes	

16 32

tilt u(32) : Specifies the tilt angle in units of (2*pi/(2^32))
error_est u(32): Specifies the estimation error in units of (2*pi/(2^32))

RollTag

Roll angle (acceleration sensor present).

length = 12 2 Byte	**tag = 3** 2 Byte
roll 4 Bytes	
error_est 4 Bytes	

16 32

roll u(32) : Specifies the roll angle in units of (2*pi/(2^32))
error_est u(32): Specifies the estimation error in units of (2*pi/(2^32))

RotateTag

Image Sensor rotation is active (if tag is missing an unrotated image has to be assumed).

length = 8 2 Byte	**tag = 4** 2 Byte
rotate 4 Bytes	

16 32

rotate u(32) : Specifies the rotate angle in units of (2*pi/(2^32)). This value can be added to the roll angle.

VectorTag

Acceleration vector (acceleration sensor present).

length = 8 2 Byte	**tag = 5** 2 Byte
x 4 Bytes	
y 4 Bytes	
z 4 Bytes	

16 32

acceleration vector is given in integer i(32) in milli g

CONF_PREDEFINED_MOUNTING_LIST

Tag code	NumDes	Message	SNMP Support
0x0c76	video line	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_user	contains predefined mounting positions separated by semicolon, last three characters defines the associated tilt angle
Write	p_octet	access_right_service	write not supported

CONF_GETPROFILE_ALGO_PRESET_NAME_LIST

Tag code	NumDes	Message	SNMP Support
0x0c6c	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	
Write	p_octet	access_right_service	write not supported

This command returns a list of VCA profile specific information. These information contains the VCA algorithm type, the associated dome preset (if existent otherwise zero) and the profile name.

1	2	4	6	42	
Num Profiles 1 Bytes	**Reserved** 1 Byte	**Algorithm type** 2 Byte	**Preset** 2 Byte	**Profile Name** 32 Byte	**...**
		Profile 1			Profile 2

Algorithm Types

Flag	Name
0x0001	Object Based VCA (IVA, EVA)
0x0002	Flow Based
0x0004	Block Based VCA (Motion)

CONF_COMMISSION_SETUP

Tag code	NumDes	Message	SNMP Support
0x0c6b	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	returns status of commissioning
Write	p_octet	access_right_service	moves camera to coordinates commissioning

Read:

Payload Structure

0 – 4	8 – 31
Flags 4 Bits	**reserved** 28 Bits

Flags

Bit 0:	1: commissioning available	0: commisioning not available
Bit 1:	1: running	0: idle
Bit 2:	1: last command failed	0: last command succeeded, will be set to 0 with each write command
Bit 3:	1: Is initialized	0: not initialized
Bit 4-31:	reserved, value read is undefined	

Write:

Payload Structure

16 – 32
Flags 4 Bytes
New view top left x 4 Bytes
New view top left y 4 Bytes
New view bottom right x 4 Bytes
New view bottom right y 4 Bytes

8 – 24

General

coordinates are relative in 1/0x8000. Corner top left is (0,0), bottom right is (0x8000,0x8000)
if new view top left coordinate is same as new view bottom right coordinate,
zoom will not be changed, just view is repositioned

Flags

Bit 0 - 31: reserved, write 0

CONF_DEFAULTS

Tag code	NumDes	Message	SNMP Support
0x093d	no	no	yes

	Datatype	Access Level	Description
Read	-	access_right_minimal	not supported
Write	flag	access_right_service	sets config to default values except IP, subnet and gateway and resets the device. ATTENTION: Wait till board has fully rebooted before switching of power, to ensure that defaults are applied completly. Also propagade this to the user, that he has to wait until the reboot is complete.

CONF_FACTORY_DEFAULTS

Tag code	NumDes	Message	SNMP Support
0x09a0	no	no	yes

	Datatype	Access Level	Description
Read	-	access_right_minimal	not supported
Write	flag	access_right_service	set the configuration to factory defaults and reset the board. ATTENTION: Wait till board has fully rebooted before switching of power, to ensure that defaults are applied completly. Also propagade this to the user, that he has to wait until the reboot is complete.

CONF_BOARD_RESET

Tag code	NumDes	Message	SNMP Support
0x0811	no	no	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	not supported
Write	flag	access_right_service	reset the board

CONF_BOOT_DEFAULT_APP

Tag code	NumDes	Message	SNMP Support
0x099f	no	no	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	not supported
Write	flag	access_right_service	boot the default application

CONF_VIDEO_ENC_PRIO

Tag code	NumDes	Message	SNMP Support
0x0a81	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	reads the video encoder priorisation 0:= no, 1:=h26x, 2:=jpeg, 3:=h26x_2nd stream
Write	t_dword	access_right_service	writes the video encoder priorisation 0:=no, 1:=h26x, 2:=jpeg, 3:=h26x_2nd stream

CONF_LED_BLINKING

Tag code	NumDes	Message	SNMP Support
0x013d	no	no	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	always 1
Write	flag	access_right_minimal	1=power LED will flash for 7 sec

CONF_JPEG

Tag code	NumDes	Message	SNMP Support
0x099e	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_live	20 byte write payload: byte1: 0=QCIF, 1=CIF, 2=CIF 3=4CIF; byte1:jpeg quality (0 is best); byte2,3 reserved; bytes 4-15: reserved, 16:=borderY, 17:=borderU, 18:= border V, 19:= borderWidth;
Write	not supported	access_right_live	-

CONF_CLUSTER_GROUP_SETTING

Tag code	NumDes	Message	SNMP Support
0x09ca	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	list of (DWORD ip, DWORD flags, OCTET[6] mac, WORD reserved) cluster members including this unit
Write	p_octet	access_right_minimal	list of (DWORD ip, DWORD flags, OCTET[6] mac, WORD reserved) cluster members including this unit

CONF_CLUSTER_ID

Tag code	NumDes	Message	SNMP Support
0x09cb	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the logical position inside a cluster (0=stand alone, >0=slot nbr.)
Write	t_dword	access_right_service	only in generic: set vrm instance (0 <= ClusterId < 32)

CONF_APP_OPTION

Tag code	NumDes	Message	SNMP Support
0x09e0	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	DEPRICATED: Use CONF_APP_OPTION_EXT. Returns a bit field containing 256 bits (32 byte). Each bit represents one active license on the device. Bit order is MSB first.
Write	void	access_right_service	not supported

CONF_APP_OPTION_EXT

Tag code	NumDes	Message	SNMP Support
0x0c4a	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Returns a bit field containing 512 bits (64 byte). Each bit represents one active license on the device. Bit order is MSB first.
Write	void	access_right_service	not supported

CONF_APP_OPTION_UNIT_ID

Tag code	NumDes	Message	SNMP Support
0x09e1	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read out the unique unit id (installation code) for setting application options
Write	void	access_right_service	not supported

CONF_APP_OPTION_SET

Tag code	NumDes	Message	SNMP Support
0x09e2	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	returns a readable string with the result code of the last applied set option
Write	p_string	access_right_service	a key, generated by the license server to enable a specific application option; after entering a key, the APP_OPTION command returns the current activated application options

CONF_CPU_LOAD_IDLE

Tag code	NumDes	Message	SNMP Support
0x0a06	host=1, coproc=2	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	cpu load idle in percent
Write	t_dword	access_right_service	not supported

CONF_CPU_LOAD_CODER

Tag code	NumDes	Message	SNMP Support
0x0a07	host=1, coproc=2	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	cpu load coder in percent
Write	t_dword	access_right_service	not supported

CONF_CPU_LOAD_VCA

	Tag code	NumDes	Message	SNMP Support
	0x0a08	host=1, coproc=2	no	yes
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	cpu load vca in percent	
Write	t_dword	access_right_service	not supported	

CONF_STORAGE_LOAD

	Tag code	NumDes	Message	SNMP Support
	0x0bba	no	no	yes
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	load of local storage in percent	
Write	t_dword	access_right_service	not supported	

CONF_CPU_LOAD

	Tag code	NumDes	Message	SNMP Support
	0x0a0a	host=1, coproc=2	no	yes
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	cpu load in percent (byte 0=idle, byte 1 = coder, byte 2=vca, byte 3 LocalRecordingLoad)	
Write	p_octet	access_right_service	not supported	

CONF_SYSTEM_LOAD

Tag code	NumDes	Message	SNMP Support
0x0cb5	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	recording bitrates:
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
length 2 Byte	**Tag** 2 Byte
Payload 1..n Bytes	

8 24

length

Length of the tag incuding the 4 header bytes

Values:

Tag

ID of parameter

Values:

Payload

Value of parameter or nested tag

Values:

Tagged format

The format has an internal tagged structure with the following top level tags:

CPU_LOAD_OF_CORE_N	1
RECORDER_N	2

The top level tag CPU_LOAD_OF_CORE_N can contain the following nested tags:

CORE_IDX	3
IDLE_PERC	5
ENC_PERC	6
VCA_PERC	7

The top level tag RECORDER_N can contain the following nested tags:

REC_IDX	4
RECORDING_LOAD_PERC	8
RECORDING_MEDIA_STATUS	9

CPU_LOAD_OF_CORE_N

Container tag which contains information about current CPU-load on a single CPU-core

length = 4..n 2 Byte	**tag = 1** 2 Byte
Nested tags 0..n Bytes	

16 32

CORE_IDX

Index of the core described by the current container

length = 5 2 Byte	**tag = 3** 2 Byte
Idx 0..n 1 Bytes	

16 32

IDLE_PERC

Idle value for this core in percent

length = 5	**tag = 5**

2 Byte	2 Byte
Percent 0..100 1 Bytes	

16 32

ENC_PERC

CPU-load in percent caused by the encoder on this core

length = 5 2 Byte	**tag = 6** 2 Byte
Percent 0..100 1 Bytes	

16 32

VCA_PERC

CPU-load in percent caused by the VCA on this core

length = 5 2 Byte	**tag = 7** 2 Byte
Percent 0..100 1 Bytes	

16 32

RECORDER_N

Container tag which contains information about a local recording media.
The availability of a index in this tag depends on the availability of a media slot on the device, and not on wether a media is currently plugged.
This means: If a device has 2 SD-Card slots then this command always returns 2 descriptors (idx 0,1) also when only one or none card is plugged.

length = 4..n 2 Byte	**tag = 2** 2 Byte
Nested tags 0..n Bytes	

16 32

REC_IDX

Index of the media slot, e.g. SD-Card, described by the current container.

length = 5 2 Byte	**tag = 4** 2 Byte
Idx 0..n 1 Bytes	

16 32

RECORDING_LOAD_PERC

Current load of a local recording media in percent

length = 5 2 Byte	**tag = 8** 2 Byte
Current load 0..100 percent 1 Bytes	

16 32

RECORDING_MEDIA_STATUS

Current status of a local recording media

length = 8 2 Byte	**tag = 9** 2 Byte
Status code 4 Bytes	

16 32

RECORDING_MEDIA_STATUS Status codes

Values:

No device plugged	1
Device detected and can be used	2
Hardware init failed	3
Hardware driver IO error	4
Hardware CRC error	5
Hardware IO timeout	6
Warning, Media has sometimes slow IO	7
Warning, Media is read-only (Either write-protected or end of life)	8
Media not suitable	9
Media too small	10

CONF_CPU_COUNT

Tag code	NumDes	Message	SNMP Support
0x0a09	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	number of cpus
Write	t_dword	access_right_service	not supported

CONF_SANITY_CHECK

Tag code	NumDes	Message	SNMP Support
0x09da	no	yes	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the array of sanity bits (bit0: hdd, bit1: device error host, bit2: device error coproc)
Write	p_octet	access_right_service	set the array of sanity bits (bit0:hdd)

CONF_ONVIF_STREAM_URI_EX

Tag code	NumDes	Message	SNMP Support
0x0be4	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	gets a extension to the onvif stream uri that will be appended in the GetStreamUri command
Write	p_string	access_right_service	sets a extension to the onvif stream uri that will be appended in the GetStreamUri command

CONF_NIGHT_MODE_STATE

Tag code	NumDes	Message	SNMP Support
0x0aa2	-	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=night mode off; 1=night mode on (only for devices supporting night mode)
Write	void	access_right_service	not supported

CONF_BLOCK_BASED_STORAGE

Tag code	NumDes	Message	SNMP Support
0x0adf	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_internal	not supported
Write	p_octet	access_right_internal	read or write 512 byte memory blocks of flash (service for 3rd party modules like imgage-pipe dll or fpga)

Payload Structure

16 32

Domain 4 Bytes
Operation 4 Bytes
PageIndex 4 Bytes
PageCount 4 Bytes
Padding 16 Bytes ...
Payload n x 512 Bytes ...

8 24

Domain

Identifies the memory area to be accessed. Each module may have its own areas. The areas and their size have to be predefined by the firmware. Some may be permanent and some may get erased by the factory reset button.

Operation

Type of operation to be performed

Values:

BBSS_OP_READ	0
BBSS_OP_WRITE	1

PageIndex

Index of the first 512 byte page to be accessed

Values:

Index	0..n-1

PageCount

Number of 512 byte pages to be accessed

Padding

Padding up to a total hdr size of 32 bytes

Values:

Should be zero	0

Payload

n * 512 bytes payload to be read or written

Description

Read or write n blocks of 512 bytes to flash. All single blocks can be simply overwritten. A explicit erase is not required. Write is atomic on block basis which means that also in case of power-loss-while-write either the consistent old version or the new version of a single block is read back next time after reboot therefore doublebuffering is not required.

CONF_POE_GRANTED_POWER

	Tag code	NumDes	Message	SNMP Support
	0x0ae6	no	no	yes

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the current poe power granted by switch / midspan in tenths of Watts
Write	t_word	access_right_service	write/limit the max power budget to be used by the device in tenths of Watts

CONF_INSTALLER_SEQUENCE

	Tag code	NumDes	Message	SNMP Support
	0x0b0f	cam	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	0: installer sequence is deactivated; 1: installer sequence is allowed
Write	t_dword	access_right_service	0: deactivate installer sequence; 1: allow installer sequence

CONF_UPLOAD_HISTORY

	Tag code	NumDes	Message	SNMP Support
	0x0b44	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the history of uploads to this device
Write	p_octet	access_right_internal	save the history of uploads to this device

CONF_STATIC_SERVER_URL

Tag code	NumDes	Message	SNMP Support
0x0b63	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	url of a static server for firmware updates (e.g. http://downloadstore.boschsecurity.com/index.php)
Write	p_string	access_right_service	url of a static server for firmware updates (e.g. http://downloadstore.boschsecurity.com/index.php)

CONF_GB28181

Tag code	NumDes	Message	SNMP Support
0x0ba2	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	configure GB28181
Write	p_octet	access_right_service	configure GB28181

Payload Structure

16	32
Enable 1 Byte	**H264 Stream Cfg** 1 Byte	**Reserved** 2 Byte
Heart beat time out 4 Bytes	
Registration time out 4 Bytes	
Server port 2 Byte	**Device Port** 2 Byte
Server URL 64 Bytes ...	
...	
Server ID 21 Bytes ...	
...	
	Device ID 21 Bytes
...	
	Alarm Device Id 21 Bytes
...	
	Password 128 Bytes
...	
	Reserved 753 Bytes
...	
8	24

Enable

1: Enable GB28181 client 0: Disable

H264 Stream Cfg

if 1, use h264 elementary stream (including startcodes) directly packed into RTP packets (non-RFC conform, but needed for GB28181 test tool)

Reserved

Must be written as 0

Heart beat time out

Define heart beat time out in seconds max. one day min. 5 seconds

Registration time out

Define registration time out in seconds max. one day min. 5 seconds

Server port

currently 5060 or 5511 can be used (server and device port must be equivalent server port is also used for device port)

Device Port

currently 5060 or 5511 can be used (server and device port must be equivalent server port is also used for device port)

Server URL

URL of GB28181 server

CONF_PRIVACY_MODE

Tag code	NumDes	Message	SNMP Support
0x0c33	no	yes	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Read the privacy mode status: Byte 0 = On(1) / Off(0), Byte 1 = Source of Privacy Mode (RCP cmd = 0, Tap = 1), Byte 2-3 = opt. Timeout in min
Write	t_dword	access_right_user	Set the privacy mode status: Byte 0 = On(1) / Off(0), Byte 1 = reserved (set to '0'), Byte 2-3 = opt. Timeout in minutes (only applicable if Byte 0 is 'On'. Then, after the timeout, privacy mode is automatically switched 'off' again)

CONF_RUN_QR_READER

Tag code	NumDes	Message	SNMP Support
0x0c3a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	Run the QR code reader for a certain time (in seconds). (notice: values 0 and 0xFFFFFFFF not allowed; reserved for future use)

CONF_PTZ_AUTO_ROTATE_MODE

Tag code	NumDes	Message	SNMP Support
0x0c3f	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	residential indoor ptz auto-rotate (auto-track) mode. 0=off, 1=std function, 2=production test mode
Write	t_octet	access_right_service	residential indoor ptz auto-rotate (auto-track) mode. 0=off, 1=std function, 2=production test mode

CONF_POWER_MONITOR_VALUES

Tag code	NumDes	Message	SNMP Support
0x0c53	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns values from all monitored power rails, see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
Number of rails 4 Bytes	
Number of rails values n x 40 Bytes	
...	
8	24

Number Of Rails

Number of monitored rails. Also indicates the number of rail values structure. Payload of complete command is 4 + (Number Of Rails)*40 Bytes

Number of rails values

structure for single rail values see below

Rail Values Structure

16	32
Options 4 Bytes	
Actual Voltage 4 Bytes	
Min Voltage 4 Bytes	
Max Voltage 4 Bytes	
Actual Current 4 Bytes	
Min Current 4 Bytes	
Max Current 4 Bytes	
Actual Power 4 Bytes	
Min Power 4 Bytes	
Max Power 4 Bytes	
8	24

Options

Values:

This is an or'ed bitfield of values listed below

current measurement valid	0x01
voltage measurement valid	0x02
power measurement valid	0x04

Actual Voltage

actual measured voltage, which is measured last, in uV

Min Voltage

minimum measured voltage in uV

Max Voltage

maximum measured voltage in uV

Actual Current

actual measured current, which is measured last in uA

Min Current

minimum measured current in uA

Max Current

maximum measured current in uA

Actual Power

actual measured power, which is measured last in uW

Min Power

minimum measured power in uW

Max Power

maximum measured power in uW

CONF_POWER_MONITOR_NAMES

Tag code	NumDes	Message	SNMP Support
0x0c54	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns names from all monitored power rails, see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16 32

Number of rails 4 Bytes
Names of all rails Number of rails names
...

8 24

Number Of Rails

Number of monitored rails.

Names of all rails

All names back to back, separated by zero

CONF_AUXILIARY_POWER

Tag code	NumDes	Message	SNMP Support
0x0c55	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns status and availability of auxiliary power, see detailed description
Write	t_dword	access_right_service	enables or disables auxiliary power, see detailed description

Payload Structure

has_pwr 1 Bit	ena_pwr 1 Bit	has_ovl 1 Bit	ovl 1 Bit	reserved 4 Bits	ovl_cnt 8 Bits	reserved 16 Bits

has_pwr

indicates if the unit has auxiliary power (read only, write will be ignored)

Values:

has no auxiliray power	0
has auxiliray power	1

ena_pwr

indicates if auxiliary power is switched on

Values:

auxiliray power off	0
auxiliray power on	1

has_ovl

indicates if the unit has overload detection (read only, write will be ignored)

Values:

has no overload detection	0
has overload detection	1

ovl

indictaes if auxiliary power is overloaded (read only, write will be ignored)

Values:

auxiliary power is not overloaded	0
auxiliary power is overloaded	1

ovl_cnt

counts up if an overload occurs

Values:

no overload	0
nr of overload events	1 - 254
more than 254 overload events	255

reserved

read is undefined, should be zero when written

CONF_BLUETOOTH_CTRL

Tag code	NumDes	Message	SNMP Support
0x0c5c	Message : no	Description : returns Scan information of Bluetooth devices ##desc:./doc/CONF_BLUETOOTH_SCAN.htm##	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	%	access_right_service	%

CONF_BLUETOOTH_SCAN

Tag code	NumDes	Message	SNMP Support
0x0c5d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns Scan information of Bluetooth devices
Write	%	access_right_service	%

Payload Structure

16 32

Scan Status	n Entries	reserved
8 Bits	1 Byte	2 Bytes

Sequence of n Entries:

Len	Status
2 Bytes	16 Bits
MAC	
4 Bytes	
MAC (cont.)	**RSSI**
2 Bytes	2 Bytes
Name	
m Bytes	

16 32

Scan Status

4 8

On/Off	Discover	Truncate	reserved
1 Bit	1 Bit	1 Bit	5 Bits

On/Off

Bluetooth switched on

Values:

Off	0
Off	1

Discover

1: Discovery is running, there will be no entries during discovery, retry later

Truncate

1: List is incomplete, not all devices listed

Entries

Number of Entries (n)

Len

Total Len of Entry

Status

reserved

MAC

Mac address of BT device

RSSI

value t.b.d

Name

Name of BT device, terminated with zero

CONF_LLDP_REQUESTED_POWER_TOTAL

Tag code	NumDes	Message	SNMP Support
0x0c8e	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the poe power requested by device in total (adder+device itself) in tenths of Watts
Write	%	access_right_service	%

CONF_LLDP_REQUESTED_POWER_CAM

Tag code	NumDes	Message	SNMP Support
0x0c8f	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the poe power requested by device (for the device alone) in tenths of Watts
Write	%	access_right_service	%

CONF_LLDP_ALLOCATED_POWER_TOTAL

Tag code	NumDes	Message	SNMP Support
0x0c90	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the poe power allocated for the device in total (adder+device itself) in tenths of Watts
Write	%	access_right_service	%

CONF_LLDP_POWER_ADDER

Tag code	NumDes	Message	SNMP Support
0x0c92	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the power volume that the camera should request via LLDP in addition to its own requirements; value is in tenths of watts; e.g. 9.9 watts results in a value of 99;
Write	t_word	access_right_service	set the power volume that the camera should request via LLDP in addition to its own requirements; value is in tenths of watts; e.g. 9.9 watts results in a value of 99; when the adder value is different from 0, the device identifies as type 2 PD of class 4; when adder value is 0 and the device specific requirement is below 13 W, the device signals type 1 PD of the respective class; currently, the total power requested by a camera can not exceed the 25.5W specified as maximum in IEEE802.3at-2009

CONF_BASE_POE_CLASS

Tag code	NumDes	Message	SNMP Support
0x0ca0	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_service	read the base poe class of the device (w/o considering power adder); 1: class0, 2: class1, 3: class2, etc. 255: no PoE
Write	%	access_right_service	%

CONF_SIGNALLED_POE_CLASS

Tag code	NumDes	Message	SNMP Support
0x0ca1	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_service	read the signalled poe class of the device (incl. power adder); 1: class0, 2: class1, 3: class2, etc. 255: no PoE
Write	%	access_right_service	%

CONF_STRATOCAST_ONOFF

Tag code	NumDes	Message	SNMP Support
0x0cab	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	Switch on or off the automatic start of stratocast service
Write	f_flag	access_right_service	Switch on (true) or off (false) the automatic start of stratocast service

CONF_STRATOCAST_REGISTER

Tag code	NumDes	Message	SNMP Support
0x0cad	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	not supported
Write	p_string	access_right_service	Register at stratocast service and switch on automatic reconnect. As payload the activation code must be provided

CONF_STRATOCAST_STATE

Tag code	NumDes	Message	SNMP Support
0x0cb4	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Get the actual state of the stratocast connection (0=inactive, 1=connecting, 2=registering, 3=connected)
Write	t_dword	access_right_service	not supported

CONF_NBR_OF_IMU

Tag code	NumDes	Message	SNMP Support
0x0cd5	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of available IMU sensors (IMU: Inertial Measurement Unit; accelerometer, gyroscope, ..)
Write	t_dword	access_right_service	not supported

CONF_MPEG_AUDIO_SAMPLING_FREQ

Tag code	NumDes	Message	SNMP Support
0x0932	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	mpeg audio sampling frequency (Hz)
Write	t_dword	access_right_service	not supported

CONF_MPEG4_CURRENT_PARAMS

	Tag code	NumDes	Message	SNMP Support
	0x0600	coder instance (1 to n, 0 for session based)	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	get the current profile preset number for the specified encoder (NumDesc)	
Write	t_dword	access_right_service	sets the video encoder (given by NumDesc) to a preset (payload). If recording is running and the iframe distance is 0, the iframe distance of the preset will be patched to a recording suitable value. For transcoder: if the request contains the replay session id and no num descriptor, the device will assign the preset to the transcoder of the replay session. If the preset is empty the transcoder will be disabled. For exclusive (dual) stream instances: if the request contains the session id and the num descriptor is 0, the previously configured preset with its session based (non-permanent) modifications will be applied to the encoder.	

CONF_MPEG4_CURRENT_PARAMS_TRANSCODER

	Tag code	NumDes	Message	SNMP Support
	0x0b4e	has to be zero	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	sessionID references the replay session; 0:= no transcoding; 1-8: video preset of the transcoder	
Write	t_dword	access_right_user	assign preset n (1,..N) to transcoder session; preset 0 means no transcoding	

CONF_MPEG4_CURRENT_PARAMS_REL_CODER

Tag code	NumDes	Message	SNMP Support
0x061c	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the current profile preset number for the specified encoder (relative to line).
Write	p_octet	access_right_service	sets the video encoder (relative to line) to a preset.

Payload Structure

16 32

Line 1 Byte	**Coder** 1 Byte	**Coding Capabilities** 2 Bytes	
Preset 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

8 24

Line

Video input line

Coder

Relative coder number (relative to line)

Coding Capabilities

Coding capabilities of encoder.
All coding capabilities are one or multiple of:

Values:

H.263	0x0002
Mpeg 4	0x0004
Mpeg 2	0x0008
H.264	0x0040

Preset

Number of the encoder preset profile

Notes:

When reading the preset profile, the according byte in the payload will be set.

CONF_MPEG4_NAME

Tag code	NumDes	Message	SNMP Support
0x0602	profile preset	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the name of the preset given by the Numeric Descriptor
Write	p_string	access_right_service	sets the name of the preset number given by the Numeric Descriptor

CONF_MPEG4_BANDWIDTH_KBPS

Tag code	NumDes	Message	SNMP Support
0x0607	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	reads the bandwidth of selected preset
Write	t_dword	access_right_service	sets the bandwidth of selected preset, sessionID to address session based video params

CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT

Tag code	NumDes	Message	SNMP Support
0x0612	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	reads the bandwidth soft limit (in KBPS) of selected preset
Write	t_dword	access_right_service	sets the bandwith soft limit (in KBPS) of selected preset, sessionID to address session based video params

CONF_MPEG4_BANDWIDTH_KBPS_HARD_LIMIT

Tag code	NumDes	Message	SNMP Support
0x0613	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete
Write	t_dword	access_right_service	obsolete

CONF_MPEG4_INTRA_FRAME_DISTANCE

Tag code	NumDes	Message	SNMP Support
0x0604	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the intra frame distance of selected preset
Write	t_dword	access_right_service	sets the intra frame distance of selected preset (not supported while recording is running or configured to active and the preset is used on the recording schedule), sessionID to address session based video params

CONF_MPEG4_FRAME_SKIP_RATIO

Tag code	NumDes	Message	SNMP Support
0x0606	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets Mpeg4 frame skip ratio (1=all frames ,...)
Write	t_dword	access_right_service	set Mpeg4 frame skip ratio (1=all frames ,...), sessionID to address session based video params

CONF_MPEG4_RESOLUTION

	Tag code	NumDes	Message	SNMP Support
	0x0608	profile preset, 0 if sessionID is used	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the spatial resolution for selected preset. Possible resolutions depending on the device platform (CPPx). They can be queried with the command CONF_ENC_PROFILE_RESOLUTION_OPTIONS. Info: for devices where CONF_ENC_BASE_OPERATION_MODE_TYPE is '1' this command only applies for SD resolutions (0=QCIF, 1=CIF, 2=2CIF, 3=4CIF, 4=(1/2 D1), 5=(2/3D1), 6=QVGA, 7=VGA, 8=WD144 (256x144), 9=WD288 (512x288), 10=WD432 (768x432), 18=WD1 (960H)). HD resolutions then need to be set via CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE. For devices where CONF_ENC_BASE_OPERATION_MODE_TYPE is '0' (no base operation modes) then this command also applies for HD resolutions. The t_dword payload is then as follows: upper 2 Bytes of DWORD = width, lower 2 Bytes = height. (e.g. 1280x720 -> t_dword = 0x050002D0 = 83886800dec); HD resolutions on transcoder: 720 (up to 720x1280), 1080 (up to 1080x1920), 2160(up to 2160x3840); SessionID to address session based video params
Write	t_dword	access_right_service	sets the spatial resolution for selected preset. Possible resolutions depending on the device platform (CPPx). They can be queried with the command CONF_ENC_PROFILE_RESOLUTION_OPTIONS. Info: for devices where CONF_ENC_BASE_OPERATION_MODE_TYPE is '1' this command only applies for SD resolutions (0=QCIF, 1=CIF, 2=2CIF, 3=4CIF, 4=(1/2 D1), 5=(2/3D1), 6=QVGA, 7=VGA, 8=WD144 (256x144), 9=WD288 (512x288), 10=WD432 (768x432), 18=WD1 (960H)). HD resolutions then need to be set via CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE. For devices where CONF_ENC_BASE_OPERATION_MODE_TYPE is '0' (no base operation modes) then this command also applies for HD resolutions. The t_dword payload is then as follows: upper 2 Bytes of DWORD = width, lower 2 Bytes = height. (e.g. 1280x720 -> t_dword = 0x050002D0 = 83886800dec); HD resolutions on transcoder: 720 (up to 720x1280), 1080 (up to 1080x1920), 2160(up to 2160x3840); SessionID to address session based video params

CONF_MPEG4_FIELD_MODE

	Tag code	NumDes	Message	SNMP Support
	0x060e	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get Mpeg4 field mode (0=progressive, 1=merged, 2=separated)
Write	t_octet	access_right_service	set Mpeg4 field mode (0=progressive, 1=merged field, 2=separated field)

CONF_MPEG4_DEFAULTS

	Tag code	NumDes	Message	SNMP Support
	0x0601	profile preset	no	no

	Datatype	Access Level	Description
Read	void	access_right_minimal	not supported
Write	flag	access_right_service	set the selected preset params to the default values (if the recording is running and the preset in on the recording schedule, the preset will keep its old iframe distance)

CONF_MPEG4_I_FRAME_QUANT

Tag code	NumDes	Message	SNMP Support
0x060a	yes	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the I frame quantizer (1-31,0=auto) for selected preset
Write	t_dword	access_right_service	sets the I frame quantizer (1-31,0=auto) of selected preset

CONF_MPEG4_P_FRAME_QUANT

Tag code	NumDes	Message	SNMP Support
0x060b	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the P frame quantizer (1-31,0=auto) for selected preset
Write	t_dword	access_right_service	sets the P frame quantizer (1-31,0=auto) for selected preset

CONF_MPEG4_PARAMS_MAX_NUM

Tag code	NumDes	Message	SNMP Support
0x0614	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Number of MPEG4 presets
Write	%	access_right_service	%

CONF_MPEG4_AVC_I_FRAME_QUANT

Tag code	NumDes	Message	SNMP Support
0x0615	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the I frame quantizer (9-51,0=auto) for selected preset
Write	t_dword	access_right_service	sets the I frame quantizer (0=auto,9-51) for selected preset

CONF_MPEG4_AVC_P_FRAME_QUANT

Tag code	NumDes	Message	SNMP Support
0x0616	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the P frame max quantizer (9-51,0=auto) for selected preset
Write	t_dword	access_right_service	sets the P frame max quantizer (0=auto,9...51) for selected preset

CONF_MPEG4_AVC_P_FRAME_QUANT_MIN

	Tag code	NumDes	Message	SNMP Support
	0x0620	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the P frame min quantizer (9-51,0=auto) for selected preset
Write	t_dword	access_right_service	sets the P frame min quantizer (0=auto,9...51) for selected preset, sessionID to address session based video params

CONF_MPEG4_AVC_DELTA_IPQUANT

	Tag code	NumDes	Message	SNMP Support
	0x0621	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the difference (-10..+10) between I and P-Frame quantization for selected preset
Write	t_int	access_right_service	sets the difference (-10..+10) between I- and P-Frame quantization for selected preset

CONF_MPEG4_AVC_QUANT_ADJ_REGION_1

	Tag code	NumDes	Message	SNMP Support
	0x0624	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the quality difference (H.264 QP Units; -51..+51) between normal and background regions; use CONF_ROI_OPTIONS to query options if only positiv offsets are allowed offsets between 0..+51 can be read
Write	t_int	access_right_service	sets the quality difference (H.264 QP units; -51..+51) between normal and background regions; use CONF_ROI_OPTIONS to query options if only positiv offsets are allowed offsets between 0..+51 can be set

CONF_MPEG4_AVC_QUANT_ADJ_REGION_2

Tag code	NumDes	Message	SNMP Support
0x0625	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the quality difference (H.264 QP Units; -51..+51) between normal and object regions; use CONF_ROI_OPTIONS to query options if only positiv offsets are allowed offsets between 0..+51 can be read
Write	t_int	access_right_service	sets the quality difference (H.264 QP units; -51..+51) between normal and object regions; use CONF_ROI_OPTIONS to query options if only positiv offsets are allowed offsets between 0..+51 can be set

CONF_MPEG4_AVC_DEBLOCKING_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0617	profile preset	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	gets the deblocking filter status for selected preset
Write	flag	access_right_service	enables/disables the H264 deblocking filter

CONF_MPEG4_AVC_DEBLOCKING_ALPHA

Tag code	NumDes	Message	SNMP Support
0x0618	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the alpha coefficient of the H264 deblocking filter for the selected preset
Write	t_int	access_right_service	set the alpha H264 deblocking coefficent (-5...5)

CONF_MPEG4_AVC_DEBLOCKING_BETA

Tag code	NumDes	Message	SNMP Support
0x0619	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the beta coefficient of the H264 deblocking filter for the selected preset
Write	t_int	access_right_service	set the beta H264 deblocking coefficent (-5...5)

CONF_MPEG4_AVC_CHROMA_QUANT_OFF

Tag code	NumDes	Message	SNMP Support
0x061a	profile preset	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	gets the chroma quantisation offset for the selected preset
Write	t_int	access_right_service	set the chroma quantisation offset (-12...12)

CONF_MPEG4_AVC_CODING_MODE

Tag code	NumDes	Message	SNMP Support
0x0a45	profile preset	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	gets the coding mode of the selected preset (0=frame; 1=field; 2=makro block adaptive ff; 3=picture adaptive ff)
Write	t_octet	access_right_service	sets the coding mode for the selected preset (0=frame; 1=field; 2=makro block adaptive ff; 3=picture adaptive ff)

CONF_MPEG4_AVC_GOP_STRUCTURE

Tag code	NumDes	Message	SNMP Support
0x0a94	profile preset	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	gets the GOP structure of the selected preset (0=IP; 1=IBP; 2=IBBP; 3=IBBRBP)
Write	t_octet	access_right_service	sets the GOP structure for the selected preset (0=IP; 1=IBP; 2=IBBP; 3=IBBRBP)

CONF_GOP_STRUCTURE_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bef	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get a list of options, which can be selected by CONF_MPEG4_AVC_GOP_STRUCTURE options are defined by MPEG4_AVC_GOP_STRUCTURE (Format: Byte 0 (len); Byte (1-len) supported options)
Write	p_octet	access_right_service	not supported

CONF_VIDEO_ENC_P_REF_LIST_SIZE

Tag code	NumDes	Message	SNMP Support
0x0627	profile preset	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	max. number of reference frames used by a P-slices in H.26x streams with 0 = unrestricted
Write	t_octet	access_right_service	max. number of reference frames used by a P-slice in H.26x streams with 0 = unrestricted. Query largest support value via GET_P_REF_LIST_SIZE_LIMIT

CONF_GET_VIDEO_ENC_P_REF_LIST_SIZE_LIMIT

Tag code	NumDes	Message	SNMP Support
0x0628		no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get the largest value supported by VIDEO_ENC_P_REF_LIST_SIZE; 0 indicates that the value is not user-adjustible
Write	t_octet	access_right_service	not supported

CONF_MPEG4_AVC_CABAC

Tag code	NumDes	Message	SNMP Support
0x0aa6	profile preset	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	gets cabac for the selected preset (0=off; 1=on)
Write	t_octet	access_right_service	sets cabac for the selected preset (0=off; 1=on)

CONF_MPEG4_AVC_BITRATE_OPTIMIZATION_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0c49	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] basicSupport; Byte[1] reserved; Byte[2-7] reserved
Write	p_octet	access_right_service	not supported

CONF_ENC_DYN_SCENE_CTRL

Tag code	NumDes	Message	SNMP Support
0x0cb6	line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] 0: disable / 1: enable; Byte[1]: 1: Stream 1 leads, 2: Stream 2 leads, 3: Stream 3 leads ... ; default 255 smart stream selection Byte[2-15] reserved; support can be queried via CONF_ENC_DYN_SCENE_CTRL_OPTIONS
Write	p_octet	access_right_service	Byte[0] 0: disable / 1: enable; Byte[1]: 0: Stream 0 leads, 1: Stream 1 leads, 2: Stream 2 leads ... ; default 255 smart stream selection Byte[2-15] reserved; support can be queried via CONF_ENC_DYN_SCENE_CTRL_OPTIONS

CONF_ENC_DYN_SCENE_CTRL_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0cb7	line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] 1: CONF_ENC_DYN_SCENE_CTRL is supported 0 not supported; Byte[1] 0: one imagepipe per line 1: separated imagepipe per stream; stream count Byte[2-15] reserved
Write	p_octet	access_right_service	Byte[0] 1: CONF_ENC_DYN_SCENE_CTRL is supported 0 not supported; Byte[1] 0: one imagepipe per line 1: separated imagepipe per stream; stream count Byte[2-15] reserved

CONF_GET_ENC_DYN_SCENE_CTRL_INFO

Tag code	NumDes	Message	SNMP Support
0x0cb8	line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

tag 2 Byte	**length** 2 Byte
payload length Bytes ...	
tag 2 Byte	**length** 2 Byte
payload length Bytes ...	

16 32

Generic description

CONF_GET_ENC_DYN_SCENE_CTRL_INFO
Every tagged command consits of an id and a length field + payload.

Tags

Image pipe offset per line info

tag = 0 2 Byte		**length = 8** 2 Byte	
leading stream 1 Byte	**current encoder profile** 1 Byte	**current bitrate reduction mode** 1 Byte	**sharpness offset** 1 Byte
temporal noise filter offset 1 Byte	**spatial noise filter offset** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

8 16 24 32

leading stream: Stream 0 leads, 1: Stream 1 leads, 2: Stream 2 leads ... ; default 255 smart stream selection
current bitrate reduction mode: CONF_MPEG4_AVC_BITRATE_OPTIMIZATION 0 = OFF, 1 = normally no visible effects (tuned for low amount of artefacts),
2 = very low amount of visable artefacts 3 = no artefacts in most scenes (best balance between bitrate and quality),
4 = some visible artefacts, but very low bitrate 5 = artefacts in many scenes but very low bitrate
sharpness offset
temporal noise filter offset
spatial noise filter offset

Tags

Image pipe offset per line info

tag = 1 2 Byte	**length = 8** 2 Byte
stream id 2 Byte	**reserved** 2 Byte
bitrate modifier 4 Byte	

8 16 24 32

stream id: 1..n
referenced stream
bitrate modifier:
in 1/1000 e.g. 900 -> 0.9 bitrate after interaction of encoder with image pipe is reduced by up to 10 %

CONF_MPEG4_AVC_BITRATE_OPTIMIZATION

	Tag code	NumDes	Message	SNMP Support
	0x0c37	profile preset	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] 0 = OFF, 1 = normally no visible effects (tuned for low amount of artefacts), 2 = very low amount of visable artefacts 3 = no artefacts in most scenes (best balance between bitrate and quality), 4 = some visible artefacts, but very low bitrate 5 = artefacts in many scenes but very low bitrate, Byte [1-7] reserved; if activated (byte 1 not 0) CONF_MPEG4_BANDWIDTH_KBPS will only have an influence if CONF_VIDEO_BITRATE_AVERAGING_PERIOD is not 0 otherwise device will be in a varibale bitrate mode (only a max bitrate can be configured via CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT
Write	p_octet	access_right_service	Byte[0] 0 = OFF, 1 = normally no visible effects (tuned for low amount of artefacts), 2 = very low amount of visable artefacts 3 = no artefacts in most scenes (best balance between bitrate and quality), 4 = some visible artefacts, but very low bitrate 5 = artefacts in many scenes but very low bitrate, Byte [1-7] reserved; if activated (byte 1 not 0) CONF_MPEG4_BANDWIDTH_KBPS will only have an influence if CONF_VIDEO_BITRATE_AVERAGING_PERIOD is not 0 otherwise device will be in a varibale bitrate mode (only a max bitrate can be configured via CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT

CONF_CODER_SPECIFIC_ENC_PROFILES

	Tag code	NumDes	Message	SNMP Support
	0x0c9b	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	0 means not supported on this platform; if supported all commands that address a encoder profile via the 'num' parameter have to send the abs video tx coder in the upper 8bits of the 'num';
Write	t_dword	access_right_service	not supported

CONF_VIDEO_BITRATE_AVERAGING_PERIOD

	Tag code	NumDes	Message	SNMP Support
	0x0622	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the (maximum) period in seconds for bitrate averaging (0=no bitrate averaging)
Write	t_dword	access_right_service	sets the (maximum) period in seconds for bitrate averaging (0=no bitrate averaging)

CONF_VIDEO_QUALITY

Tag code	NumDes	Message	SNMP Support
0x0a82	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the video quality for a selected preset (0=auto, 1(worst)..100(best))
Write	t_dword	access_right_service	sets the video quality for a selected preset (0=auto, 1(worst)..100(best))

CONF_JPEG_BANDWIDTH_KBPS

Tag code	NumDes	Message	SNMP Support
0x061d	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the jpeg bandwidth (in KBPS) of selected preset
Write	t_dword	access_right_service	sets the bandwith for jpeg streaming

CONF_JPEG_BANDWIDTH_KBPS_SOFT_LIMIT

Tag code	NumDes	Message	SNMP Support
0x061e	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the jpeg bandwidth soft limit(in KBPS) of selected preset
Write	t_dword	access_right_service	sets the bandwith soft limit for jpeg streaming

CONF_JPEG_BANDWIDTH_KBPS_HARD_LIMIT

Tag code	NumDes	Message	SNMP Support
0x061f	profile preset	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the jpeg bandwidth hard limit(in KBPS) of selected preset
Write	t_dword	access_right_service	sets the bandwith hard limit for jpeg streaming

CONF_VID_ENCODER_ON

Tag code	NumDes	Message	SNMP Support
0x0262	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	=1: video encoder is enabled on establishing a new connection; =0 video encoder is not enabled on establishing a new connection
Write	flag	access_right_service	=1 enables the video encoder on establishing a new connection; =0 disables the video encoder on establishing a new connection

CONF_VIDEO_ENCODER_STATUS

Tag code	NumDes	Message	SNMP Support
0x09df	coder instance	no	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get encoder status (8bytes) 4 (high order) bytes 10*frames per sec, 4 (low order) bytes KBPS
Write	p_octet	access_right_service	not supported

CONF_VIDEO_ENCODER_STATUS_EXT

Tag code	NumDes	Message	SNMP Support
0x0a90	coder instance	no	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get encoder status (20 bytes) 1st DWORD: 10*frames per sec, 2nd DWORD: KBPS, 3rd DWORD: grab frame fps*10, 4th lost frame fps*10 due to overload, 5th DWORD skipped frame fps*10 due to configured enc interval in profile or STREAM_PRIORITY
Write	p_octet	access_right_service	not supported

CONF_EXT_ENCODER_BITRATE_STATISTICS

Tag code	NumDes	Message	SNMP Support
0x0c85	coder instance	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	not supported

Payload Structure

GlobalParameters.localTimeSecoundsSince2000 4 Bytes
GlobalParameters.UTCSecoundsSince2000 4 Bytes
GlobalParameters.reserved 24 Bytes
Record[0].timeBaseInSec 4 Bytes
Record[0].lastCellIsActiveSinceSec 4 Bytes
Record[0].elementCount 4 Bytes
Record[0].reserved 20 Bytes
Record[0].bitrate[Record[0].elementCount] 4 Bytes * Record[0].elementCount
Record[1].timeBaseInSec 4 Bytes
Record[1].lastCellIsActiveSinceSec 4 Bytes
Record[1].elementCount 4 Bytes
Record[1].reserved 20 Bytes
Record[1].bitrate[Record[1].elementCount] 4 Bytes * Record[1].elementCount
Record[2] ... 4 Bytes * Record[2].elementCount+32
Record[n] 4 Bytes * Record[n].elementCount+32

16 32

Generic description

The command provides enhanced encoder bitrate statistics every record describes a table of bitrates.
The table has a time base in sec typically sec, min, hour, days and weeks is provided
The first entry of the list can also only be active since a part of the cell e.g. cell time base is one day
may this is only active since 300 sec. This time is signaled via lastCellIsActiveSinceSec

Definition of the global parameters, these parameters are vaild for all record entrys:
GlobalParameters.localTimeSecoundsSince2000 local camera time which corresponds to the delivered entrys
GlobalParameters.UTCSecoundsSince2000 utc time which corresponds to the delivered entrys

Definition of the record entrys:
timeBaseInSec : base time for all cells 2-elementCount base time of cell 1 is signaled via lastCellIsActiveSinceSec
lastCellIsActiveSinceSec : entry [0] is active since lastCellIsActiveSinceSec
elementCount : nr elements
bitrate[1...elementCount]: example timeBaseInSec = 60, lastCellIsActiveSinceSec = 20 sec, elementCount = 256
bitrate[1] : average bitrate{0 s ,...- 20 s}
bitrate[2] : average bitrate{-20 s,...- 80 s}
bitrate[3] : average bitrate{-80 s,...- 140 s}
bitrate[...]
bitrate[elementCount]

CONF_VIDEO_ENCODER_ENCODED_BYTES

Tag code	NumDes	Message	SNMP Support
0x0af6	coder instance	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Get encoded bytes (32 Bit Counter) since bootup. Only supported for h.264 and Jpeg encoder.
Write	t_dword	access_right_service	not supported

CONF_ENC_CURRENT_RESOLUTION

Tag code	NumDes	Message	SNMP Support
0x0b4b	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the current resolution (width + height) in pixel of an encoder (relative to line; starting with '1'). First two bytes of payload (line+coder) need to be specified as request payload. 1st Byte: video line (needs to be set when reading); 2nd Byte: encoder (needs to be set when reading); 3.+4. Byte: width; 5.+6. Byte: height; 7.+8. Byte reserved
Write	p_octet	access_right_service	not supported

CONF_ENC_PROFILE_BASIC_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0b2e	profile preset	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get basic parameters of an encoder profile preset. 1st DWORD: Resolution (parameter values see CONF_MPEG4_RESOLUTION); 2nd DWORD: Skip Ratio (values see CONF_MPEG4_FRAME_SKIP_RATIO); 3rd DWORD: Target bit rate (values see CONF_MPEG4_BANDWIDTH_KBPS); 4th DWORD: Maximum bit rate (values see CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT)
Write	p_octet	access_right_service	set basic parameters of an encoder profile preset and apply them instantly. 1st DWORD: Resolution (parameter values see CONF_MPEG4_RESOLUTION); 2nd DWORD: Skip Ratio (values see CONF_MPEG4_FRAME_SKIP_RATIO); 3rd DWORD: Target bit rate (values see CONF_MPEG4_BANDWIDTH_KBPS); 4th DWORD: Maximum bit rate (values see CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT)

CONF_ENC_PROFILE_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0cb9	profile preset	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the parameters of an encoder profile preset. Reply is in tagged format, see detailed description
Write	p_octet	access_right_service	set one or more parameters of an encoder profile preset. Payload is in tagged format, see detailed description

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes - 4	
length 2 Byte	**tag** 2 Byte
payload length Bytes - 4	

16 32

Generic description

The format has a tagged structure.
Each tagged entry consists of a length field, a tag ID and a payload.
Allows to read/write all relevant parameters of an encoder profile preset like resolution, skip, bandwidth, etc... within one single command.
In read direction all available tags are returned.
In write direction just the parameters/tags which should be changed need to be sent. So either one tag, several tags or all available tags can be provided.

Tags

Each tag ID in this command represents the according RCP command which already exists to read/write one of the encoder profile parameters individually.
I.e. this command combines several individual RCP commands within one command and provides the parameters as separate tags.
The payload of the tag entries corresponds to the related RCP command. For detailed information please check the documentation of the according RCP command.

E.g. one of the supported tags is 0x0608 (CONF_MPEG4_RESOLUTION).

length = 8 2 Byte	**tag = 0x0608** 2 Byte
resolution 4 Bytes	

8 16 24 32

The currently supported tag IDs are:

0x0602 - CONF_MPEG4_NAME
0x0604 - CONF_MPEG4_INTRA_FRAME_DISTANCE
0x0606 - CONF_MPEG4_FRAME_SKIP_RATIO
0x0607 - CONF_MPEG4_BANDWIDTH_KBPS
0x0608 - CONF_MPEG4_RESOLUTION
0x0612 - CONF_MPEG4_BANDWIDTH_KBPS_SOFT_LIMIT
0x0620 - CONF_MPEG4_AVC_P_FRAME_QUANT_MIN
0x0621 - CONF_MPEG4_AVC_DELTA_IPQUANT
0x0622 - CONF_VIDEO_BITRATE_AVERAGING_PERIOD
0x0624 - CONF_MPEG4_AVC_QUANT_ADJ_REGION_1
0x0625 - CONF_MPEG4_AVC_QUANT_ADJ_REGION_2
0x0627 - CONF_VIDEO_ENC_P_REF_LIST_SIZE
0x0a94 - CONF_MPEG4_AVC_GOP_STRUCTURE
0x0c37 - CONF_MPEG4_AVC_BITRATE_OPTIMIZATION

CONF_ROI

	Tag code	NumDes	Message	SNMP Support
	0x0b48	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_live	see detailed description

Select region of interest. The Session-ID identifies the video coder.

Payload Structure

16	32
hPos 2 Bytes	**vPos** 2 Bytes
size 2 Bytes	**reserved** 2 Bytes

hPos / vPos

Center position of the cropping window in relative coordinates (0..32768). E.g. (0, 0) is upper left; (16384, 16384) is the center of the original image.

size

Size of the cropping window in relative coordinates (0..32768). The aspect ratio will be preserved.

CONF_ROI_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0c7e	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get ROI Encoding options per Line Byte 0, [0x1 support ROI in h.264, 0x2 support ROI in h.265], Byte 1 support CONF_ROI_PRESET_POS_INTERNAL, Byte 2, [0 = default, 0x1 only allow positiv qp offsets at encoder regions are allowed; reduction o 3-7 reserved
Write	p_octet	access_right_live	not supported

CONF_VIDEO_ENC_TEMP_QUANT_ADJ

Tag code	NumDes	Message	SNMP Support
0x0626	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get temporary encoding quality adjustment see
Write	p_octet	access_right_live	temporary encoding quality adjustmens see

Payload Structure

16 32

e	deltaQp	reserved
1	7 Bits	24 Bits

left	right
16 Bits	16 Bits

top	bottom
16 Bits	16 Bits

8 24

enable

Values:

enable temporary quality adjustments	1
disable temporary quality adjustments	0

deltaQp

Intended quality adjustment in H.264 QP units. If enable is 0, this field must also be set to 0. Otherwise the allowed range is -51..51 with negative values indicating higher quality and positive ones lower quality

reserved

set to 0

left

relative position of the left edge of the adjustment region within the complete encoded image

right

relative position of the right edge of the adjustment region within the complete encoded image

top

relative position of the upper edge of the adjustment region within the complete encoded image

bottom

relative position of the lower edge of the adjustment region within the complete encoded image

Description

The encoder, to which the quality adjustment is to be applied is identified via the RCP session.

The area, in which the qualtiy is adjusted is characterized by the position of the edges of a rectangle within a virtual coordinate system with a nominal range of [0..+32768]x[0..32768] for the complete encoded picture (0=top/left edge, 32768=bottom/right edge). The actually adjusted area will be enlarged as needed by the encoder.

Note, that the payload may contaion only first octet, if enable is set to 0

CONF_DPTZ_DYNAMIC_CAPS

Tag code	NumDes	Message	SNMP Support
0x0bb7	0 - addressing via Session ID	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read current/dynamic DPTZ capabilities. 1st bit: ROI, 2nd bit: E-PTZ-Tracker
Write	t_dword	access_right_service	not supported

CONF_DPTZ_STATIC_CAPS

	Tag code	NumDes	Message	SNMP Support
	0x0bfd	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	see detailed description	
Write	%	access_right_service	%	

Payload Structure

Generic description

Read current DPTZ capabilities. They may are changed if CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE or CONF_VIDEO_LINE_DEWARPING_MODE is changed.
The command consists of n entries with a fixed length of 8 Bytes and the following data.

Tags

Record:

Line 1 Byte	**Stream** 1 Byte	**Bicom PTZ Steering** 1 Byte	**Auto Tracker Type** 1 Byte
Steering Type 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
8	16	24	32

Line: 1,2...
Stream: 1,2...
Bicom PTZ Steering: 1 PTZ commands are supported / 0 PTZ commands are not supported
Auto Tracker Type:
0 not supported
1 EPTZ supported
Steering Type:
ptzSteeringTypeNoOptionPresent = 0,
ptzSteeringTypePhysicalDome = 1,
ptzSteeringTypeCutOutRegion = 2,
ptzSteeringTypeVirtualPtz = 3

CONF_FLIP_MIRROR

	Tag code	NumDes	Message	SNMP Support
	0x0bc3	no	no	no
	Datatype	**Access Level**	**Description**	
Read	%	access_right_live	%	
Write	p_octet	access_right_live	sessionID references the transcoder session; byte 0-3 flip bits, bytes 4-7 mirror bits	

CONF_VCA_SHAPES

	Tag code	NumDes	Message	SNMP Support
	0x0bc8	abs Coder (without sessionID)	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_live	sessionID references the transcoder session; enable vca shape overlay	
Write	t_dword	access_right_live	sessionID references the transcoder session; enable vca shape overlay;	

CONF_STAMP

Tag code	NumDes	Message	SNMP Support
0x0c71	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

Stamping Region ID 2 Bytes	**Region Payload Length** 2 Bytes
Tag ID 2 Bytes	**Tag Payload Length** 2 Bytes
Tag Payload	

Use CONF_ENC_STAMPING_PROPERTIES to query basic options. Read of CONF_STAMP will return all available Tags. Depending on the device different regions are available

Stamping Region ID

ID	Description	Support visibility Tag (1)	Support position Tag (2)	Text (3)	Font size (4)	Transparent Background (5)	Color (6)
0	All regions (global settings)	NO	NO	NO	yes	yes	yes
1	Time Stamp Region	yes	yes Position Type {0,1,2,3}	NO	yes	yes	yes
2	Name Stamping Region	yes	yes Position Type {0,1,2,3}	use CONF_CAMNAME_LINES	yes	yes	yes
3	Alarm Stamping Region	use CONF_ALARM_DISP_VAL	use CONF_STAMP_ATTR_ALARM*	use CONF_ALARM_STRING	use CONF_STAMP_ATTR_ALARM	use CONF_STAMP_ATTR_ALARM	yes
4	Info Stamping Region	yes	yes Position Type {0,3}	NO	yes	yes	yes
5	Spinner Region	yes	yes Position Type {0,3}	NO	yes	yes	yes
6	System Controller Region 1	yes	yes Position Type {0,3}	internal	yes	yes	yes
7	System Controller Region 2	yes	yes Position Type {0,3}	internal	yes	yes	yes
8	System Controller Region 3	yes	yes Position Type {0,3}	internal	yes	yes	yes
9	Seperate Logo Area	yes	yes Position Type {0,3}	NO	yes	yes	no

CONF_STAMP_ATTR_ALARM* for custom position config CONF_ALARM_DISP_VAL needs to be set to 3

Region Payload Length

Number of bytes of all enclosed tags inside this region.

Tag ID

1	Visibility
2	Position
3	Text
4	Font size (only allowed in region 0)
5	Transparent Background
6	Color

Tag Payload Length

Payload length in bytes of the tag denoted in "Tag ID".

Tag Payload Length

Payload of the tag. See below for details of the respective tag.

Tag 1: Visibility

Tag ID = 1 2 Bytes	**Tag Payload Length = 1** 2 Bytes
Visibility 1 Byte	

Visibility

0	Hidden

1	Visible

Tag 2: Position

Tag ID = 2 2 Bytes		Tag Payload Length = 3 2 Bytes	
Position Type 1 Byte	**Position X** 1 Byte	**Position Y** 1 Byte	

Position Type

0 = OFF; 3 = Custom; in some commands also 1 = bottom, 2 = top is supported is this case Position X anx Y is ignored. See above table.

Position X

Position X is an integer value from 0 to 255, where 0 is the left border of the image and 255 is the right border of the image.

Position Y

Position Y is an integer value from 0 to 255, where 0 is the top border of the image and 255 is the bottom border of the image.

Tag 4: Font size

Tag ID = 4 2 Bytes		Tag Payload Length = 1 2 Bytes
Font size 1 Byte		

Font size

0	Normal
1	Big

Tag 5: Transparent Background

Tag ID = 5 2 Bytes		Tag Payload Length = 1 2 Bytes
Transparent background 1 Byte		

Transparent background

0	No transparent background
1	Transparent background

Tag 6: Color

Tag ID = 6 2 Bytes		Tag Payload Length = 12 2 Bytes	
Color Mode 1 Byte	**Reserved** 3 Bytes		
Background - Red 1 Byte	**Background - Green** 1 Byte	**Background - Blue** 1 Byte	**Reserved** 1 Byte
Text - Red 1 Byte	**Text - Green** 1 Byte	**Text - Blue** 1 Byte	**Reserved** 1 Byte

Color Mode

0: RGB values are used

Background - Red/Green/Blue

Components of background color. Integer values from 0 to 255.

Text - Red/Green/Blue

Components of text color. Integer values from 0 to 255.

CONF_ENC_STAMPING_PROPERTIES

Tag code	NumDes	Message	SNMP Support
0x0bb3	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	not supported

Payload Structure

UTF16CharsPerLine 2 Bytes		**CamNameLines** 2 Bytes	
AlarmStampingLines 1 Byte	**TimeStampSupport** 1 Byte	**TimeStampRes** 1 Byte	**TranspStampSupport** 1 Byte
bannerModeSupport 1 Byte	**extendedSysconAreas** 1 Byte	**SpinnerStampSupport** 1 Byte	**BigFontSupport** 1 Byte
seperateLogoAreaSupport 1 Byte	**infoStampSupport** 1 Byte	**maxLogoSize** 1 Byte	**reserved** 1 Byte
colorSupport 1 Byte	**reserved** 3 Bytes		

UTF16CharsPerLine

UTF16 chars per line

CamNameLines

Number of stamping lines available at CONF_CAMNAME_LINES

AlarmStampingLines

Number of alarm stamping lines

TimeStampSupport

1: time stamping support

TimeStampRes

Time stamp resolution, 1 also ms stamping is supported

TranspStampSupport

1: transparent stamping is supported

bannerModeSupport

1 : name stamping supports banner mode 0xFF : all stamping areas supports banner mode

extendedSysconAreas

Number of extended syscon areas. In CPP7.3 this is usually 3, in earlier platforms 0.

SpinnerStampSupport

1: spinner stamping support

BigFontSupport

1: big font support

seperateLogoAreaSupport

1: seperate logo area is supported

maxLogoSize

maximum logo size
1: 128x128 pixels
2: 300x300 pixels

colorSupport

1: stamping color can be changed

CONF_NAME_STAMP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x0084	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.	
Write	t_octet	access_right_user	Deprecated! Use CONF_STAMP instead.	

CONF_TIME_STAMP_VAL

	Tag code	NumDes	Message	SNMP Support
	0x0085	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.	
Write	t_octet	access_right_user	Deprecated! Use CONF_STAMP instead.	

CONF_TIME_STAMP_RESOLUTION

	Tag code	NumDes	Message	SNMP Support
	0x0a7f	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	0=second, 1=ms	
Write	t_octet	access_right_user	0=second, 1=ms	

CONF_ALARM_DISP_VAL

Tag code	NumDes	Message	SNMP Support
0x008e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	t_octet	access_right_service	1=disable alarm display; 2=enable alarm display; 3=alarm stamping with custom attributes which can be set with the CONF_STAMP_ATTR_ALARM tag

CONF_ALARM_STRING

Tag code	NumDes	Message	SNMP Support
0x0090	no	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	get the alarm string
Write	p_unicode (max 32 unicode characters)	access_right_service	set the alarm string

CONF_INFO_STAMP_VAL

Tag code	NumDes	Message	SNMP Support
0x0bc0	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	t_octet	access_right_user	Deprecated! Use CONF_STAMP instead.

CONF_LOGO_STAMP_VAL

Tag code	NumDes	Message	SNMP Support
0x0c10	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	t_octet	access_right_user	Deprecated! Use CONF_STAMP instead.

CONF_STAMP_ATTR_NAME

Tag code	NumDes	Message	SNMP Support
0x0936	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

Supported for CPP4, CPP6, CPP7 and CPP7.3. For other platforms use CONF_STAMP instead.

Payload Structure

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

x

X-Position (0...255) where the string should be displayed (0 is left)

y

Y-Position (0...255) where the string should be displayed (0 is up)

attributes (use CONF_ENC_STAMPING_PROPERTIES to determine supported options)

Values:

Bit 20	Draw a box around the text or transparent text background (0=box, 1=transparent)
Bit 21	Draw a banner from top to end of name text if y pos =< 128 from bottom if ypos > 128)

flags

Values:

Bit 0	Reserved
Bit 1	Enable logo stamping (0=off, 1=on)
Bit 2	Logo position relative to font (0=left, 1=right)
Bit 3	Display only logo (0=off, 1=on)

reserved

Reserved fields must be written as 0

CONF_STAMP_ATTR_TIME

Tag code	NumDes	Message	SNMP Support
0x0937	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	p_octet	access_right_service	Deprecated! Use CONF_STAMP instead.

CONF_STAMP_ATTR_ALARM

Tag code	NumDes	Message	SNMP Support
0x0938	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

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

x

X-Position (0...255) where the string should be displayed (0 is left)

y

Y-Position (0...255) where the string should be displayed (0 is up)

attributes (use CONF_ENC_STAMPING_PROPERTIES to determine supported options)

Values:

Bit 20	Draw a box around the text or transparent text background (0=box, 1=transparent)

flags

Values:

reserved

Reserved fields must be written as 0

CONF_STAMP_ATTR_INFO

Tag code	NumDes	Message	SNMP Support
0x0bc1	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	p_octet	access_right_service	Deprecated! Use CONF_STAMP instead.

CONF_STAMP_ATTR_LOGO

Tag code	NumDes	Message	SNMP Support
0x0c0f	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Deprecated! Use CONF_STAMP instead.
Write	p_octet	access_right_service	Deprecated! Use CONF_STAMP instead.

CONF_STAMP_WIDTH

Tag code	NumDes	Message	SNMP Support
0x0a93	video line (1,..)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the width of the current stamping in pixels. See detailed description
Write	void	access_right_service	not supported

Payload Structure

16 32

Stamping Type 2 Byte	**Reserved** 2 Byte
Width 4 Byte	

8 24

Note: This command can only be read. The payload defines the stamping which should be read and returns the width in the corresponding field.

Stamping Type

Defines the stamping type of which the width should be determined.
The type can be one of the following:

Values:

Time Stamp	0x0001
Name Stamp row 1	0x0002
Name Stamp row 2	0x0003
Alarm Stamp	0x0004
Info Stamp	0x0005

Width

Width of the stamping in pixels (hex)

CONF_VIDEO_CURRENT_PARAMS_CODNBR

Tag code	NumDes	Message	SNMP Support
0x0982	coder instance (1 to n, 0 for session based)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the current profile preset number for the specified encoder (NumDesc)
Write	t_dword	access_right_service	sets the video encoder (given by NumDesc) to a preset (payload). If recording is running and the iframe distance is 0, the iframe distance of the preset will be patched to a recording suitable value. For transcoder: if the request contains the replay session id and no num descriptor, the device will assign the preset to the transcoder of the replay session. If the preset is empty the transcoder will be disabled. For exclusive (dual) stream instances: if the request contains the session id and the num descriptor is 0, the previously configured preset with its session based (non-permanent) modifications will be applied to the encoder.

CONF_MAX_NBR_OF_ENC_STREAMS

Tag code	NumDes	Message	SNMP Support
0x029e	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Number of encoder streams
Write	t_dword	access_right_service	read only

CONF_WATERMARK

Tag code	NumDes	Message	SNMP Support
0x0924	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	0=watermark is disabled; 1=classic watermark is enabled, 2=video authentication(low profile), 3=video authentication(mid profile), 4=video authentication(high profile)
Write	t_octet	access_right_service	0=disables watermark; 1=classic watermark, 2=video authentication(low profile), 3=video authentication(mid profile), 4=video authentication(high profile)

CONF_CODER_VIDEO_OPERATION_MODE

	Tag code	NumDes	Message	SNMP Support
	0x0a9c	absolute coder instance, if zero - read out, if supported global mode	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	see detailed description
Write	t_dword	access_right_service	see detailed description

Global device configuration mode

There are two different modes signaled via INDIVIDUAL_ENCODER_OPERATION_MODE_CONFIG @CONF_DEVICE_CAPABILITIES
1: **invidual encoder configuration mode:**
CONF_CODER_VIDEO_OPERATION_MODE, num parameter for a specific absolute coder has to be provided; global configration (numDesc = 0) is not supported
0: **global encoder configuration mode :**
CONF_CODER_VIDEO_OPERATION_MODE, num parameter supports only 0 as global option for all possible encoders

line dependent or global capabilitys

1: if **invidual encoder configuration mode** is supported
use CONF_CODER_VIDEO_OPERATION_MODE_OPTIONS with NumDesc = absolute coder number to get the supported options at CONF_CODER_VIDEO_OPERATION_MODE
0: if **global encoder configuration mode** is supported
use CONF_CODER_VIDEO_OPERATION_MODE_OPTIONS with NumDesc = 0 to get the supported global options at CONF_CODER_VIDEO_OPERATION_MODE

CONF_CODER_VIDEO_OPERATION_MODE

Set one video coder (absolute coder number given by NumDesc) or all video coders, which can handle the option, depending on the supported **configuration mode**, to a video operation mode.

Allowed options can be queried by CONF_CODER_VIDEO_OPERATION_MODE_OPTIONS: 0 = jpeg, 1 = h263, 2 = h264, 3 = h265, 0x103 = h_265 without B-Frames.
Live connections will be dropped; Only applicable if recording isn't active and configured to off.

CONF_CODER_VIDEO_OPERATION_OPTION

Tag code	NumDes	Message	SNMP Support
0x0aa4	coder instance	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete: use VIDEO_H264_ENC_BASE_OPERATION_MODE instead
Write	t_dword	access_right_service	obsolete: use VIDEO_H264_ENC_BASE_OPERATION_MODE instead

CONF_CODER_VIDEO_OPERATION_MODE_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0ca3	absolute coder instance, if 0 global options are provided	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	list of options (DWORDS) which can be written at CONF_CODER_VIDEO_OPERATION_MODE for a specific coder
Write	p_octet	access_right_service	not supported

CONF_VIDEO_H264_ENC_CONFIG

Tag code	NumDes	Message	SNMP Support
0x0ad2	Message : no	Description : get the profile for each h.264 encoder per line for each time. DWORD 1: Mode, 0=live mode, 1-10: schedule number; DWORD 2: profile for first stream, DWORD 3: profile for second stream...	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	p_octet	access_right_service	set the profile for each h.264 encoder per line for each time. DWORD 1: Mode, 0=live mode, 1-10: schedule number; DWORD 2: profile for first stream, DWORD 3: profile for second stream... (if recording on that cam isn't running, the setting for live mode(0) will take place immediately); Notice: the number of expected streams can be queried with CONF_NBR_OF_ENC_STREAMS

CONF_VIDEO_H264_ENC_CONFIG_DEFAULTS

	Tag code	NumDes	Message	SNMP Support
	0x0b4d	cam	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	read not supported	
Write	t_dword	access_right_service	set the default profile for each h.264 encoder per line for each time. Payload: Mode, 0=live mode, 1-10: recording schedule number. (if recording on that cam isn't running, the setting for live mode(0) will take place imidiatly)	

CONF_VIDEO_H264_ENC_CONFIG_BULK

	Tag code	NumDes	Message	SNMP Support
	0x0ad9	Message : no	Description : get the profiles for each h.264 encoder per line for each time; this request will supply all Modes 0-10 in one reply; returns: DWORD 1: NbrOfStreamsPerMode, remaining payload 11x payload of VIDEO_H264_ENC_CONFIG	no
	Datatype	**Access Level**	**Description**	
Read	%	access_right_minimal	%	
Write	p_octet	access_right_service	set the profiles for each h.264 encoder per line for each time; this request must supply all Modes 0-10 in one request; DWORD 1: NbrOfStreamsPerMode, remaining payload 11x payload of VIDEO_H264_ENC_CONFIG; Notice: the expected 'NbrOfStreamsPerMode' can be queried with CONF_NBR_OF_ENC_STREAMS	

CONF_ENC_BASE_OPERATION_MODE_TYPE

Tag code	NumDes	Message	SNMP Support
0x0c95	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the type of base operation mode which is used on the device; 0=no base operation mode support, 1=base operation modes define a specific fixed resolution (except of SD mode)
Write	%	access_right_service	%

CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE

Tag code	NumDes	Message	SNMP Support
0x0ad3	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the base operation mode of the h.264 encoders per line. first DWORD stream 1, second DWORD stream 2...;bit 24 signals that width and height need to be exchanged (picture is rotated by 90 degrees) 0=copy other stream, 1=compatibility mode to H2.64 BP+ (bitrate limited), 3: h.264 MP SD, 4: h.264 MP 720p, 5: h.264 MP 720p full frame rate, 6: h.264 MP 1080p, 7: h264 MP 720p skip=3, 8: h264 MP 720p skip=4, 9: h264 MP 1080p skip=7, 10: h.264 MP SD ROI PTZ, 11: h.264 MP HD 2592x1944, 12: h.264 MP 400x720 upright (cropped), 13: h.264 MP SD 4CIF resolution 4:3 format (cropped), 14: h.264 MP SD dual stream with independent ROI PTZ, 15: h.264 MP HD 1280x960 4:3 (cropped), 16: h.264 MP HD 1440x1080, 17: h.264 MP HD 1280x1024, 18: h.264 MP 576x1024 upright (cropped), 19: h.264 MP HD 720p (cropped), 20: h.264 MP HD 2704x2032, 21: h.264 MP HD 2992x1680, 22: h.264 MP HD 3840x2160, 23: h.264 MP HD 4000x3000, 24: h.264 MP HD 3584x2016, 25: h.264 MP HD 800x600, 26: h.264 MP HD 1024x768, 27: h.264 MP HD 1280x960, 28: h.264 MP HD 1600x1200, 29: h.264 MP HD 3648x2160, 30: h.264 MP HD 2640x2640, 31: h.264 MP HD 1792x1792, 32: h.264 MP HD 1024x1024, 33: h.264 MP HD 800x800, 34: h.264 MP HD 1536x864, 35: h.264 MP HD 2560x1440, 36: h.264 MP HD 2560x960, 37: H264_FEDC_MODE_MP_3648x1080, 38: h.264 MP HD 1824x1080 with ROI PTZ, 39: h.264 MP HD 1824x540, 40: h.264 MP HD 1280x480, 41: h.264 MP HD 1536x1536, 42: h.264 MP HD 800x800 with ROI PTZ, 43: h.264 MP 768x768, 44: h.264 MP 768x288, 45: h.264 MP 2048x1152, 46: h.264 MP 2688x800, 47: h.264 MP 672x200, 48: h.264 MP 608x360, 49: h.264 MP 640x240, 50: h.264 MP 768x576, 51: h.264 MP 3584x504, 52: h.264 MP SD VGA (cropped), 53: h.264 2992x1680 @ SKIP 3, 54: h.264 2704x2032 @ SKIP 3, 55: h.264 2640x2640 @ SKIP 6, 56: h.264 1792x1792 @ SKIP 2, 57: h.264 1824x1080, 58: h.264 1216x720, 59: h.264 1280x1024 5:4 (cropped), 60: h.264 2048x1536, 61: h.264 1920x1080 @ SKIP 2, 62: h.264 1920x1080 (cropped), 63: h.264 3072x1728, 64: h.264 3072x1728 @ SKIP 3, 65: h.264 2688x1512, 66: h.264 2304x1296, 67: h.264 2688x1512 @ SKIP 2, 68: h.264 2304x1296 @ SKIP 2, 69: SD dual stream, 70: 3264x1840, 71: 3264x1840 @ SKIP 3, 72: dual stream 480x480 + 1024x1024, 73: dual stream 1280x720 + 640x360, 74: 1920x1080 @ SKIP2 + 176x120; 75: 3840x2160 @ SKIP6 76: 3840x2160 @ SKIP3; 77: 3584x2016 @ SKIP8; 78: 3264_1840 @ SKIP3; 79: 3072_1728 @ SKIP2; 80: 3584x2016 @ SKIP6;81: 3584x2016 @ SKIP2;82: 1920x1440_CROP; 83: 2720x1530; 84: 2720x1530 @ SKIP2; 85: 1440x1440 , query supported resolutions via CONF_H264_ENC_BASE_OP_MODE_CAPS_VERBOSE set up the base operation mode of the h.264 encoders per line. first DWORD stream 1, second DWORD stream 2...;bit 24 signals that width and height need to be exchanged (picture is rotated by 90 degrees), 0=copy other stream, 1=compatibility mode to H2.64 BP+ (bitrate limited), 3: h.264 MP SD, 4: h.264 MP 720p, 5: h.264 MP 720p full frame rate, 6: h.264 MP 1080p, 7: h264 MP 720p skip=3, 8: h264 MP 720p skip=4, 9: h264 MP 1080p skip=7, 10: h.264 MP SD ROI PTZ, 11: h.264 MP HD 2592x1944, 12: h.264 MP 400x720 upright (cropped), 13: h.264 MP SD 4CIF resolution 4:3 format (cropped), 14: h.264 MP SD dual stream with independent ROI PTZ, 15: h.264 MP HD 1280x960 4:3 (cropped), 16: h.264 MP HD 1440x1080, 17: h.264 MP HD 1280x1024, 18: h.264 MP 576x1024

Write	p_octet	access_right_service	upright (cropped), 19: h.264 MP HD 720p (cropped), 20: h.264 MP HD 2704x2032, 21: h.264 MP HD 2992x1680, 22: h.264 MP HD 3840x2160, 23: h.264 MP HD 4000x3000, 24: h.264 MP HD 3584x2016, 25: h.264 MP HD 800x600, 26: h.264 MP HD 1024x768, 27: h.264 MP HD 1280x960, 28: h.264 MP HD 1600x1200, 29: h.264 MP HD 3648x2160, 30: h.264 MP HD 2640x2640, 31: h.264 MP HD 1792x1792, 32: h.264 MP HD 1024x1024, 33: h.264 MP HD 800x800, 34: h.264 MP HD 1536x864, 35: h.264 MP HD 2560x1440, 36: h.264 MP HD 2560x960, 37: H264_FEDC_MODE_MP_3648x1080, 38: h.264 MP HD 1824x1080 with ROI PTZ, 39: h.264 MP HD 1824x540, 40: h.264 MP HD 1280x480, 41: h.264 MP HD 1536x1536, 42: h.264 MP HD 800x800 with ROI PTZ, 43: h.264 MP 768x768, 44: h.264 MP 768x288, 45: h.264 MP 2048x1152, 46: h.264 MP 2688x800, 47: h.264 MP 672x200, 48: h.264 MP 608x360, 49: h.264 MP 640x240, 50: h.264 MP 768x576, 51: h.264 MP 3584x504, 52: h.264 MP SD VGA (cropped), 53: h.264 2992x1680 @ SKIP 3, 54: h.264 2704x2032 @ SKIP 3, 55: h.264 2640x2640 @ SKIP 6, 56: h.264 1792x1792 @ SKIP 2, 57: h.264 1824x1080, 58: h.264 1216x720, 59: h.264 1280x1024 5:4 (cropped), 60: h.264 2048x1536, 61: h.264 1920x1080 @ SKIP 2, 62: h.264 1920x1080 (cropped), 63: h.264 3072x1728, 64: h.264 3072x1728 @ SKIP 3, 65: h.264 2688x1512, 66: h.264 2304x1296, 67: h.264 2688x1512 @ SKIP 2, 68: h.264 2304x1296 @ SKIP 2, 69: SD dual stream, 70: 3264x1840, 71: 3264x1840 @ SKIP 3, 72: dual stream 480x480 + 1024x1024, 73: dual stream 1280x720 + 640x360, 74: 1920x1080 @ SKIP2 + 176x120; 75: 3840x2160 @ SKIP6 76: 3840x2160 @ SKIP3; 77: 3584x2016 @ SKIP8; 78: 3264_1840 @ SKIP3; 79: 3072_1728 @ SKIP2; 80: 3584x2016 @ SKIP6;81: 3584x2016 @ SKIP2;82: 1920x1440_CROP; 83: 2720x1530; 84: 2720x1530 @ SKIP2; 85: 1440x1440 , query supported resolutions via CONF_H264_ENC_BASE_OP_MODE_CAPS_VERBOSE

CONF_VID_H264_ENC_BASE_OPERATION_MODE_CAPS

	Tag code	NumDes	Message	SNMP Support
	0x0af9	line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	get the base operation mode capability list of the h.264 encoders per line. The list contains all possible mode combinations for all streams. 1st DWORD is number of streams (n). Then the combinations follow paired in n DWORDS. Possible modes: see CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE	
Write	p_octet	access_right_service	not supported	

CONF_VIDEO_H264_ENC_CURRENT_PROFILE

Tag code	NumDes	Message	SNMP Support
0x0ad4	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns the current active profile number for the h.264 streams (first DWORD: stream 1, second DWORD: stream 2...)
Write	-	access_right_service	not supported

CONF_JPEG_STREAM_SETUP

Tag code	NumDes	Message	SNMP Support
0x0ad5	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns the configuration of the JPEG encoder; resolution defined by the id may vary please use CONF_JPEG_STREAM_SETUP_OPTIONS_VERBOSE to query which ids are supported and there properties DWORD 1:Id, DWORD 2: fps in mHz, DWORD 3: quality (0=auto, 1(worst)..100(best))
Write	p_octet	access_right_service	write the configuration of the JPEG encoder; resolution defined by the id may vary please use CONF_JPEG_STREAM_SETUP_OPTIONS_VERBOSE to query which ids are supported DWORD 1: Id, DWORD 2: fps in mHz, DWORD 3: quality (0=auto, 1(worst)..100(best))

CONF_JPEG_STREAM_SETUP_OPTIONS_VERBOSE

Tag code	NumDes	Message	SNMP Support
0x0c00	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	%	access_right_service	%

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure.
Every tagged command consits of an id and a length field + payload. Only read is supported allows to query information about the ids and the allowed ids configurable via CONF_JPEG_STREAM_SETUP.

Tags

Basic verbose of CONF_JPEG_STREAM_SETUP_OPTIONS_VERBOSE (must always be present): 0

Map CONF_JPEG_STREAM_SETUPids to jpeg width and height

length = 20 2 Byte	**tag = 0** 2 Byte
id 4 Bytes	
width 4 Bytes	
height 4 Bytes	
reserved 4 Bytes	

8 16 24 32

id: id supported in CONF_JPEG_STREAM_SETUP command
width: jpeg width, 0 -> device automaticaly decides which resolution delivers the best performance in conjuction with the current encoder configuration
height: jpeg height, 0 -> device automaticaly decides which resolution delivers the best performance in conjuction with the current encoder configuration
reserved:

CONF_JPEG_STREAM_FRAME_RATES

Tag code	NumDes	Message	SNMP Support
0x0c81	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	%	access_right_service	%

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure.
Every tagged command consits of an id and a length field + payload. Only read is supported.

Tags

Global frame rates, independent from the resoltion: 1

A proposed list of max. frame rates to be used as selection for the max. frame rate of the JPEG stream.

length = 4 + 4N 2 Byte	**tag = 1** 2 Byte
1. Frame rate in mHz 4 Bytes	
... 4 Bytes	
N. Frame rate in mHz 4 Bytes	

8 16 24 32

CONF_ENC_PROFILE_RESOLUTION_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0c99	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	%	access_right_service	%

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes - 4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes - 4...	

16 32

Generic description

The format has a tagged structure.
Each tagged command consits of a length field a tag id + payload. Only read is supported.
Allows to query information about the resolution IDs which can be used in an encoder profile, set via CONF_MPEG4_RESOLUTION.

Tags

Basic tag of CONF_ENC_PROFILE_RESOLUTION_OPTIONS (must always be present): 1

Obtain all allowed CONF_MPEG4_RESOLUTION IDs together with their according width and height

length = 20 2 Byte	**tag = 1** 2 Byte
id 4 Bytes	
width 4 Bytes	
height 4 Bytes	
reserved 4 Bytes	

8 16 24 32

id: id supported by the CONF_MPEG4_RESOLUTION command
width: according width, 0 -> device automaticaly decides which resolution delivers the best performance in conjuction with the current encoder configuration
height: according height, 0 -> device automaticaly decides which resolution delivers the best performance in conjuction with the current encoder configuration
reserved:

CONF_TCP_RATE_CONTROL

Tag code	NumDes	Message	SNMP Support
0x0b4c	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	enables/disables tcp rate control
Write	t_dword	access_right_service	enables/disables tcp rate control

CONF_TCP_BANDWIDTH_CHECK

Tag code	NumDes	Message	SNMP Support
0x0b64	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	get tcp bandwith check max duration in ms, sessionID to address session based params
Write	t_dword	access_right_user	set tcp bandwith check max duration in ms, sessionID to address session based params

CONF_TCP_BANDWIDTH_CHECK_RESULT

Tag code	NumDes	Message	SNMP Support
0x0b66	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	get result of tcp_bandwidth_check (session id required), 4bytes error, 4bytes kbps
Write	p_octet	access_right_minimal	not supported

CONF_SEI_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0b6d	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	read SEI configuration from transcoder replay connections; session ID is required; 0=off, 1=on
Write	t_octet	access_right_user	configure SEI for transcoder replay connections; session ID is required; 0=off, 1=on

CONF_PIC_INFO

Tag code	NumDes	Message	SNMP Support
0x0b8e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	enables/disables the generation of pic info packets (payload type 97)
Write	t_octet	access_right_service	enables/disables the generation of pic info packets (payload type 97)

CONF_IMAGE_STABILIZATION_MODE

Tag code	NumDes	Message	SNMP Support
0x0c01	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	enable image stabilization 0=off, 1=upscale, 2=boxing (transcoder session id required)
Write	t_dword	access_right_user	enable image stabilization 0=off, 1=upscale, 2=boxing (transcoder session id required)

CONF_TC_PLUGIN_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0c05	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	gets tc plugin params (transcoder session id required)
Write	p_octet	access_right_user	sets tc plugin params (transcoder session id required)

CONF_STREAM_PRIORITY

Tag code	NumDes	Message	SNMP Support
0x0cb0	line (1...n)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the h26x stream (1..n) which should have high priority. Set to 0 if all encoder should run free
Write	t_dword	access_right_service	select the h26x streams (1..n) which should have high priority, by setting the enc inst bit (1<

CONF_NBR_OF_ENC_STREAMS

Tag code	NumDes	Message	SNMP Support
0x0cb1	line (1...n)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the number of supported individual h26x streams
Write	%	access_right_service	%

CONF_REQ_FAST_UPDATE

Tag code	NumDes	Message	SNMP Support
0x01d3	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	not supported
Write	flag	access_right_minimal	request a video intra frame, SessionID is mandatory

CONF_MPEG4_INTRA_FRAME_REQUEST

Tag code	NumDes	Message	SNMP Support
0x0605	no	no	no

	Datatype	Access Level	Description
Read	void	access_right_minimal	not supported
Write	t_dword	access_right_minimal	request a video intra frame, SessionID is mandatory

CONF_CAMERA_LENS_CURVE

Tag code	NumDes	Message	SNMP Support
0x0be6	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	not supported

Payload Structure

length	tag
2 Byte	2 Byte
payload	
length Bytes -4	

16 32

Generic description

The command allows to query the lens transfer function of the camera.
This command is not supported for all cameras. In the case of CONF_DEVICE_CAPABILITIES -> PTZ_ON_CLIENT_TAG = TRUE
A lens curve must be present.

```
Coordinate System [Picture]h
(x = 0,y = 0)      ------------------------------------ (x = 0, y = 32768)
...
...                        (x = 16384, y = 16384)
...
(x = 0,y = 32768) ------------------------------------ (x = 32768,y = 32768)
```

Tags

Optical center: 0

Defines the horizontal and vertical position of the optical axis of the lens within the sensor plane (full viewed image)
(called image center below) in the normalized coordinate system defined in the previous description.
An offset of (16384; 16384) therefore means that the optical center of the circular projection is horizontally and vertically centered in the sensor image (should be the normal case).

length = 4+n	tag = 0
2 Byte	2 Byte
x;y	
n Bytes	

16 32

x,y (string: 5;5): specify the position of the optical center x;y
x E 0 ... 32768
y E 0 ... 32768

Example:
x;y = "16384;16384"
-> x = 16384, y = 16384 -> lens is centered

Tags

Rotation symmetric transfer function: 1

The lens curve is assumed to be rotation symmetric

length = 4+n	tag = 1
2 Byte	2 Byte
radius[0];angle[0];radius[1];angle[1]...	
n Bytes	

16 32

lens transfer function:
Specifices a tuple for a set of points (radius on the sensor,
the same coordinate system is used as before and a angle of
incidence relativ to the optical axis)

Optimal interpolation between the points can be achived with natural splines

lens transfer function (string: 5;3.6;5;3.6 ...):
lens transfer function = "7189;20.469667;13815;40.049053;17223;50.729707;20151;60.518720;21646;65.859047"
-> p[0] = (radius = 7189, angle of incidence = 20.469667°)
-> p[1] = (radius = 13815, angle of incidence = 40.049053°)
-> p[2] = (radius = 17223, angle of incidence = 50.729707°)
-> p[3] = (radius = 20151, angle of incidence = 60.518720°)
-> p[4] = (radius = 21646, angle of incidence = 65.859047°)

-> for a radius bigger than the last point the behavoiur of the lens is undefined (should be masked out in black on the client)

CONF_LOW_LIGHT_MIN_FPS

Tag code	NumDes	Message	SNMP Support
0x0ad8	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	min allowed frame rate for video sensor if low light mode is active value in mHz
Write	t_dword	access_right_service	min allowed frame rate for video sensor if low light mode is active value in mHz

CONF_VID_IN_CONTRAST

Tag code	NumDes	Message	SNMP Support
0x092b	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	reads the contrast value (0-255)
Write	t_octet	access_right_service	sets the contrast value (0-255)

CONF_VID_IN_SATURATION

Tag code	NumDes	Message	SNMP Support
0x092c	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	reads the saturation value (0-255)
Write	t_octet	access_right_service	sets the saturation value (0-255)

CONF_VID_IN_BRIGHTNESS

Tag code	NumDes	Message	SNMP Support
0x092a	video line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	reads the brightness value (0-255)
Write	t_octet	access_right_service	sets the brightness value (0-255)

CONF_VID_IN_SAMPLING_MODE

Tag code	NumDes	Message	SNMP Support
0x0aa3	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete
Write	p_octet	access_right_service	obsolete

CONF_VID_IN_WHITE_BALANCE_MODE

Tag code	NumDes	Message	SNMP Support
0x0aa8	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	0=Auto White Balance,1=Manual White Balance actual RGB-Gain values are used, see CONF_VID_IN_RGB_GAIN
Write	t_dword	access_right_service	0=Auto White Balance; 1=Manual White Balance actual RGB-Gain values are used, see CONF_VID_IN_RGB_GAIN; 2=Hold AWB actual RGB-Gain values are saved (only possible, if AWB-Mode is active), they can be accessed by CONF_VID_IN_RGB_GAIN

CONF_VID_IN_RGB_GAIN

	Tag code	NumDes	Message	SNMP Support
	0x0aa9	video line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	3 bytes: byte 0 =Red gain (0-255); byte 1 =Green gain (0-255), byte 2 =Blue gain (0-255)	
Write	p_octet	access_right_service	3 bytes: byte 0 =Red gain (0-255); byte 1 =Green gain (0-255), byte 2 =Blue gain (0-255)	

CONF_VID_IN_MIRROR

	Tag code	NumDes	Message	SNMP Support
	0x0aaa	video line	no	no
	Datatype	**Access Level**	**Description**	
Read	flag	access_right_minimal	0=deactivate image mirror; 1=activate image mirror	
Write	flag	access_right_service	0=deactivate image mirror; 1=activate image mirror	

CONF_VIDEO_TERMINATION_RESISTOR_ON

	Tag code	NumDes	Message	SNMP Support
	0x0274	video line	no	no
	Datatype	**Access Level**	**Description**	
Read	flag	access_right_minimal	state of the analog video termination resistor (75 Ohms on/off)	
Write	flag	access_right_service	switch the analog video termination resistor (75 Ohms on/off)	

CONF_VIDEO_TERMINATION_RESISTOR_OFF

Tag code	NumDes	Message	SNMP Support
0x0275	video line	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	state of the analog video termination resistor (75 Ohms off/on)
Write	flag	access_right_service	switch the analog video termination resistor (75 Ohms off/on)

CONF_VIDEO_INPUT_FORMAT

Tag code	NumDes	Message	SNMP Support
0x0504	video line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete to be removed
Write	t_dword	access_right_service	obsolete to be removed

CONF_VIDEO_INPUT_FORMAT_EX

Tag code	NumDes	Message	SNMP Support
0x0b10	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	

Payload Structure

Video Format Mode 1 Byte	**Video Format Id** 1 Byte	**Video Format Image orientation** 1 Byte	**Reserved** 1 Byte
rotation low byte 1 Byte	**rotation high byte** 1 Byte	**rotation mode** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

16 32

use CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE to query valid Options:
Video Format Mode:
0= Fixed, 1=Autodetect; signals during read if device is in auto state -> 1 device is in auto mode; active id will be shown in Video Format Id
1 = Autodetect can be set if CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE includes id =0
Video Format Id:
valid ids can be queried with CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE

Video Format Image orientation:
rotate 0° no mirror = 0
rotate 0° mirror = 1
rotate 90° no mirror = 2
rotate 90° mirror = 3
rotate 180° no mirror = 4
rotate 180° mirror = 5
rotate 270° no mirror = 6
rotate 270° mirror = 7

rotation low/high byte:
2 bytes rotation angle 0 -> 0xFFFF

rotation mode:
0=standard, 1=crop, 2=zoom

Example:
CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE

length = 20 2 Byte		**tag = 0** 2 Byte	
id = 1 1 Byte	**crop = 0** 1 Byte	**rotationOptions = 0xFF** 1 Byte	**reserved** 1 Byte
width = 1920 4 Bytes			
height = 1080 4 Bytes			
framerate_mHz = 30000 4 Bytes			

8 16 24 32

length = 24 2 Byte		**tag = 0** 2 Byte	
id = 2 1 Byte	**crop = 0** 1 Byte	**rotationOptions = 0x01** 1 Byte	**reserved** 1 Byte
width = 1280 4 Bytes			
height = 720 4 Bytes			
framerate_mHz = 60000 4 Bytes			
framerateMin_mHz = 15000 4 Bytes			

8 16 24 32

length = 20 2 Byte		**tag = 0** 2 Byte	
id = 0	**crop = 0**	**rotationOptions =**	**reserved = 0**

1 Byte	1 Byte	**0** 1 Byte	1 Byte
width = 0 4 Bytes			
height = 0 4 Bytes			
framerate_mHz = 0 4 Bytes			

8 16 24 32

CONF_VIDEO_INPUT_FORMAT_EX:

Option 0:

Video Format Mode = 0 1 Byte	**Video Format Id = 1** 1 Byte	**Video Format Image orientation = [0,1,2,3,4,5,6,7]** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

16 32

width = 1920
height = 1080
framerate_mHz = 30000
e.g Video Format Image orientation = 3 -> rotate 90° mirror

Option 1:

Video Format Mode = 0 1 Byte	**Video Format Id = 2** 1 Byte	**Video Format Image orientation = [0]** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

16 32

width = 1280
height = 720
framerate_mHz = 60000
only Video Format Image orientation = 0 allowed -> rotate 0°

Option 2:

Video Format Mode = 1 1 Byte	**Video Format Id = 0 ignored** 1 Byte	**Video Format Image orientation = [0] ignored** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

16 32

response or read payload

Video Format Mode = 1 auto 1 Byte	**Video Format Id = 2** 1 Byte	**Video Format Image orientation = [0]** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte
Reserved 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

16 32

width = 1280
height = 720
framerate_mHz = 60000
Format Image orientation = 0 -> rotate 0°

CONF_VIDEO_INPUT_FORMAT_EX_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0b9b	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete to be removed
Write	p_octet	access_right_service	not supported

CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE

Tag code	NumDes	Message	SNMP Support
0x0bfb	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	not supported

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure.
Every tagged command consits of an id and a length field + payload. Only read is supported allows to query information about the ids present in CONF_VIDEO_INPUT_FORMAT_EX.

Tags

Basic verbose of CONF_VIDEO_INPUT_FORMAT_EX (must always be present): 0

Map CONF_VIDEO_INPUT_FORMAT_EX ids to width, height and framerate. The length of this tag may increase in future versions.

length 2 Byte		**tag = 0** 2 Byte	
id 1 Byte	**additional info bits** 1 Byte	**rotationOptions** 1 Byte	**reserved** 1 Byte
width 4 Bytes			
height 4 Bytes			
framerate_mHz 4 Bytes			
optional: framerateMin_mHz 4 Bytes			

8 16 24 32

id: 0..255: all available ids can be set via CONF_VIDEO_INPUT_FORMAT_EX defines (Video Format Id Byte [2])
id = 0, special id to signal availability of auto mode
additional 8 info bits:
0 bit (defines if the picture is croped) 0 or 1: if the crop is set to one the view will be changed when switching between different resolutions takes place
1-3 bit nr of HDR exposure-1 0-> 1 exposure,1-> 2 exposure ...
4 bit: defines if custom rotation is supported (rotation low and heigh byte in CONF_VIDEO_INPUT_FORMAT_EX) 5-7 bit reserved
width: maximum available picture width for encoding
height: maximum available picture height for encoding
framerate: maximum available framerate for encoding
rotationOptions:
rotate 0° no mirror = 0,
rotate 0° mirror = 1,
rotate 90° no mirror = 2,
rotate 90° mirror = 3,
rotate 180° no mirror = 4,
rotate 180° mirror = 5,
rotate 270° no mirror = 6,
rotate 270° mirror = 7
8 bits
0 bit = 1 -> rotate 0° no mirror is supported
1 bit = 1 -> rotate 0° mirror is supported
2 bit = 1 -> rotate 90° no mirror is supported
...

Example:
rotationOptions = 0x44
-> bit 2 and bit 6 are available (of bit 0...7)
-> rotate 90° no mirror = 2 and rotate 270° no mirror = 6 are supported

CONF_H264_ENC_BASE_OP_MODE_CAPS_VERBOSE

Tag code	NumDes	Message	SNMP Support
0x0c7c	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	not supported

Payload Structure

stream count 4 Bytes			
encBaseModeId [DESCRIPTOR 0 stream 0] 4 Bytes			
height 4 Bytes			
width 4 Bytes			
copy 1 Byte	**sd** 1 Byte	**sdFramerateDependent** 1 Byte	**crop** 1 Byte
skip 1 Byte	**roi** 1 Byte	**ptz** 1 Byte	**dualStream** 1 Byte
exclusive 1 Byte	**reserved** 3 Bytes		
reserved 1 4 Bytes			
reserved 2 4 Bytes			
reserved 3 4 Bytes			
encBaseModeId [DESCRIPTOR 0 stream 1] [e.g. max 1 if stream count = 2]] 4 Bytes			
height 4 Bytes			
width 4 Bytes			
copy 1 Byte	**sd** 1 Byte	**sdFramerateDependent** 1 Byte	**crop** 1 Byte
skip 1 Byte	**roi** 1 Byte	**ptz** 1 Byte	**dualStream** 1 Byte
exclusive 1 Byte	**reserved** 3 Bytes		
reserved 1 4 Bytes			
reserved 2 4 Bytes			
reserved 3 4 Bytes			
[DESCRIPTOR 1 stream [0] 36 Bytes			
[DESCRIPTOR 1 stream [1] [e.g. max 1 if stream count = 2]] 36 Bytes			
[DESCRIPTOR ... stream[0]] 36 Bytes			
[DESCRIPTOR ... stream[1] [e.g. max 1 if stream count = 2]] 36 Bytes			
[DESCRIPTOR N stream[0]] 36 Bytes			
[DESCRIPTOR N stream[1]] 36 Bytes			

encBaseModeId: id used in CONF_VIDEO_H264_ENC_BASE_OPERATION_MODE
height: video format height of base mode with encBaseModeId 0 -> height is not fixed
width: video format width of base mode with encBaseModeId 0 -> width is not fixed
copy: base mode is copy of an other stream
sd: base mode is sd, base mode resolution is determined by encoder profile
sdFramerateDependent: base mode resolution depends on frame rate
crop: base mode is a croped view
skip: base mode is frame rate reduced (applys a skip on the full frame rate)
roi: base mode is a region of interest stream typically ptz is supported
ptz: base mode allows ptz
dualStream: base mode supports "dual stream" two encoders are used for the stream (e.g. 2 different ptz views are possible)
exclusive: only used in combination with 'dualStream' property. Indicates if the additional stream of the 'dual stream' is for exclusive access (only one client at a time) or not.

CONF_VIDEO_INPUT_DESCRIPTION

Tag code	NumDes	Message	SNMP Support
0x0c78	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] 0: optical video, 1: thermal video; reserved Bytes[1-15]
Write	p_octet	access_right_service	not supported

CONF_INPUT_SOURCE_VAL

Tag code	NumDes	Message	SNMP Support
0x0086	yes (line)	no	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	=1: camera; =2: vcr; 3=color_plane; 4= auto
Write	t_octet	access_right_service	=1: camera; =2: vcr; =3: color_plane; =4 auto

CONF_NBR_OF_VIDEO_IN

Tag code	NumDes	Message	SNMP Support
0x01d6	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed video inputs
Write	void	access_right_minimal	not supported

CONF_PRIV_MSK

Tag code	NumDes	Message	SNMP Support
0x0ab7	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	legacy command please use CONF_PRIV_MSK_POLY
Write	p_octet	access_right_service	legacy command please use CONF_PRIV_MSK_POLY

CONF_PRIV_MSK_POLY

Tag code	NumDes	Message	SNMP Support
0x0bd8	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	configure polygon privacy mask

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_PRIV_MSK_OPTIONS options.
Every tagged command consits of an id and a length field + payload.

Tags

Color id: 0

configure the priv mask color only supported if CONF_PRIV_MSK_OPTIONS allows the color configuration

length = 8 2 Byte		**tag = 0** 2 Byte	
color mode 1 Byte	**Red** 1 Byte	**Green** 1 Byte	**Blue** 1 Byte

8 16 24 32

color mode
0: RGB values are used
1: auto color mode-> priv mask color is auto selected from the surrounding R,G,B values are ignored (use CONF_PRIV_MSK_OPTIONS to query support)
2: off (use CONF_PRIV_MSK_OPTIONS to query support)
3: pixelized (use CONF_PRIV_MSK_OPTIONS to query support)
4-n: reseserved for future modes
Red: 0...255
Green: 0...255
Blue: 0...255

Tags

Polygon (fixed camera) description id: 1

Configure up to n polygons query n via CONF_PRIV_MSK_OPTIONS.
CONF_PRIV_MSK_OPTIONS also defines if only rectangles are supported then only rectangles are allow to be send with this command.
Currently only a block configuration is supported all polygons are send at once.
Currently only non self intersecting polygons are supported refer to CONF_PRIV_MSK_OPTIONS.

length = nrPoints*4+12 2 Byte		**tag = 1** 2 Byte	
enable 1 Byte	**id** 1 Byte	**nrPoints** 1 Byte	**reserved** 1 Byte
reserved 4 Bytes			
vertex[0].x 2 Byte		**vertex[0].y** 2 Byte	
vertex[1].x 2 Byte		**vertex[1].y** 2 Byte	
... 2 Byte		**...** 2 Byte	
vertex[nrPoints-1].x 2 Byte		**vertex[nrPoints-1].y** 2 Byte	

enable: 0: privacy mask is not shown 1: privacy mask is shown
id: 0... max number privacy masks (defined via CONF_PRIV_MSK_OPTIONS)
nrPoints: 0... max number points (defined via CONF_PRIV_MSK_OPTIONS)
reserved must be written as 0

vertex[i].x: 0...32768
vertex[i].y: 0...32768

Definition of the coordinate system for privacy masking (stick to sei information)
Upper left edge : (0,0)
Upper right edge: (32768,0)
Lower left edge : (0,32768)
Lower right edge: (32768,32768)

Tags

use CONF_PRIV_MSK_OPTIONS to query if supported (CONF_PRIV_MSK_OPTIONS: Byte[7] rcp priv mask support @ dome): 2

Configure up to n polygons query n via CONF_PRIV_MSK_OPTIONS.
CONF_PRIV_MSK_OPTIONS also defines if only rectangles are supported then only rectangles are allow to be send with this command.
Only one polygon is allowed to be send at once. (One poly relates in the dome case to one position configured via CONF_PRIV_MASK_DOME_PTZ_POS
Currently only non self intersecting polygons are supported refer to CONF_PRIV_MSK_OPTIONS.

length = nrPoints*4+12 2 Byte		**tag = 2** 2 Byte	
enable 1 Byte	**id** 1 Byte	**nrPoints** 1 Byte	**zoomThreshold** 1 Byte
reserved 4 Bytes			
vertex[0].x 2 Byte		**vertex[0].y** 2 Byte	
vertex[1].x 2 Byte		**vertex[1].y** 2 Byte	
... 2 Byte		**...** 2 Byte	
vertex[nrPoints-1].x 2 Byte		**vertex[nrPoints-1].y** 2 Byte	

enable: 0: privacy mask is not shown 1: privacy mask is shown
id: 0... max number privacy masks (defined via CONF_PRIV_MSK_OPTIONS)
nrPoints: 0... max number points (defined via CONF_PRIV_MSK_OPTIONS)
zoomThreshold 0, 1 (availability defined via CONF_PRIV_MSK_OPTIONS)
reserved must be written as 0

vertex[i].x: 0...32768
vertex[i].y: 0...32768

Definition of the coordinate system for privacy masking (stick to sei information)
Upper left edge : (0,0)
Upper right edge: (32768,0)
Lower left edge : (0,32768)
Lower right edge: (32768,32768)

Tags

use CONF_PRIV_MSK_OPTIONS to query if supported (CONF_PRIV_MSK_OPTIONS: Byte[7] rcp priv mask support @ dome) global dome options id: 3

global priv mask options at dome

length = 12 2 Byte		**tag = 3** 2 Byte	
disableMasks 1 Byte	**IVABehindMask** 1 Byte	**privMaskEnlargement** 1 Byte	**reserved** 1 Byte
reserved 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
8	16	24	32

reserved must be written as 0
disableMasks: disable all priv masks
IVABehindMask: enable IVA behind privacy masks

Tags

Defines on moving cameras the view for a certain priv mask. Only for internal use! Automatically generated when a polygon is written on a dome.

length = 24 2 Byte		**tag = 4** 2 Byte
enable 1 Byte	**id** 1 Byte	**res** 2 Byte
panPosition 4 Byte		
tiltPosition 4 Byte		
zoomPosition 4 Byte		
reserved 4 Byte		
8	16	24 32

enable: 0: privacy mask is not shown 1: privacy mask is shown
id: 0... max number privacy masks (defined via CONF_PRIV_MSK_OPTIONS)
panPosition: pan position
tiltPosition: tilt position
zoomPosition: zoom position

CONF_PRIV_MASK_DOME_PTZ_POS

Tag code	NumDes	Message	SNMP Support
0x0c48	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	

Payload Structure

length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	
length 2 Byte	**tag** 2 Byte
payload length Bytes -4...	

16 32

Generic description

The format has an internal tagged structure, the allowed tags can be queried via CONF_PRIV_MSK_OPTIONS options.
Every tagged command consits of an id and a length field + payload.

Tags

priv mask dome ptz cmd: 0

priv mask dome ptz

length = 12 2 Byte		**tag = 0** 2 Byte	
command 1 Byte	**maskId** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte
reserved 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

8 16 24 32

command:
moveDomeToDefinedPos = 0 move dome to position stored at maskId
storeCurrentDomePos = 1 associate id defined in maskId to current dome position
maskId
use CONF_PRIV_MSK_OPTIONS to query number of supported mask; @CONF_PRIV_MSK_OPTIONS Byte[4] maxNrOfPrivMasks per Line
define maskID which is used in conjunction with the command

CONF_PRIV_MSK_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0bd7	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_service	write not supported

Payload Structure

colorSupport 1 Byte	**rectangleSupport** 1 Byte	**polygonSupport** 1 Byte	**maxNrOfVertices** 1 Byte
maxNrOfPrivMasks 1 Byte	**onlyLegacyCmdSupport** 1 Byte	**onlyLegacyDomeCmdSupportViaBicom** 1 Byte	**rcpDomeSupport** 1 Byte
rcpDomeZoomThresholdSupport 1 Byte	**nrPrivMskLines** 1 Byte	**domePrivMaskEnlargementSupport** 1 Byte	**IVABehindPrivMaskSupport** 1 Byte
featureBits 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

Byte[0] colorSupport [0 no, 1 yes definition of color via (R,G,B) is supported];
Byte[1] rectangleSupport [0 no, 1 yes];
Byte[2] polygonSupport [0 no, 1 yes but non self intersecting];
Byte[3] maxNrOfVertices maximum points for polygon definition;
Byte[4] maxNrOfPrivMasks per Line;
Byte[5] onlyLegacyCmdSupport
Byte[6] onlyLegacyDomeCmdSupportViaBicom
Byte[7] rcp priv mask support @ dome;
Byte[8] dome zoom threshold support;
Byte[9] nrPrivMskLines;
Byte[10] dome priv mask enlargement support;
Byte[11] IVABehindPrivMaskSupport
Byte[12] supported feature bits
bit 0 : auto color mode
bit 1 : priv mask off
bit 2 : pixelized
bit 3-7: reserved

CONF_STATIC_PRIV_MSK_OVERLAY

Tag code	NumDes	Message	SNMP Support
0x0cd3		no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	

Payload Structure

cross hair = 0 2 Bytes		**length = 8** 2 Bytes	
on = 1 / off = 0 1 Bytes	**reserved** 1 Bytes	**reserved** 1 Bytes	**reserved** 1 Bytes
reserved 1 Bytes	**reserved** 1 Bytes	**reserved** 1 Bytes	**reserved** 1 Bytes

Use CONF_STATIC_PRIV_MSK_OVERLAY_OPTIONS to query supported options.
Color is currently defined by CONF_PRIV_MSK_POLY. Currently no independent cross hair color is supported.

CONF_STATIC_PRIV_MSK_OVERLAY_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0cd9	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Byte[0] 0: cross hair not supported / 1: basic support of cross hair ; Byte[1-15] reserved;
Write	%	access_right_service	%

CONF_VIDEO_STATIC_SCENE_REGIONS

Tag code	NumDes	Message	SNMP Support
0x0623	line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	static encoding categories see
Write	p_octet	access_right_service	static encoding categories see

Payload Structure

The actual payload of this command consist of a sequence of variable length records, each specifying a region and a corresponding category to be used during encoding. The region is characterized by a mandatory rectangle and optionally a shape. Up to 8 regions and in total up to 96 32-bit data for shapes are supported. Depending on the firmware version the shape data may be ignored. In this case the rectangle is used instead. This may also happen, if the restrictions indicated below are violated by the shape data.

Normally the "background" category is used for unimportant parts of the scene and the "object" category for important ones. For this two categories, specific quality adjustments can be defined via the encoding profile of every encoder. An area not belonging to any region is implicitly assigned to the "Default" category with no specific adjustments. The forth category is currently assigned via the VIDEO_ENC_TEMP_QUANT_ADJ command separately for each encoder.

Note, that by using overlapping regions, more than one category can be assigned to some areas of the scene. In this case the category with the best quality adjustment for an encoder takes preceedence. Therefore it may be usefull to assign the "Default" category explicitly to some regions.

Record Structure

		16					32
Preset id only valid for a Dome 8 Bits	**reserved** 6 Bits	**ex** 2	**reserved** 5 Bits	**ca** 2	**s** 1	**r** 1	**N** 7 Bits
left edge 16 Bits			**right edge** 16 Bits				
upper edge 16 Bits			**lower edge** 16 Bits				
hPos of vertex 1 16 Bits			**vPos of vertex 1** 16 Bits				
hPos of vertex 2 16 Bits			**vPos of vertex 2** 16 Bits				
hPos of vertex 3 16 Bits			**vPos of vertex 3** 16 Bits				
...							
hPos of vertex N 16 Bits			**vPos of vertex N** 16 Bits				
8			24				

reserved

set to 0

exclude

This bitmask indicates for which encoders this area shall NOT be used. The bits are allocated according to the the relative coder number starting with the rightmost bit (LSB) for the relative coder number 1.

reserved

set to 0

category

By default the complete scene is assumed to belong to the "default" category.

Values:

Default	0
Background	1
Objects	2
Reserved	3

shrink

Depending on the encoding resolution, the quality may only be adjusted on quite a coarse grid. This bit allows to specifiy whether the required rounding shall be performed by shrinking or enlarging the specified region.

Values:

enlarge as needed	0
shrink as needed	1

reserved

set to 0

N

This field indicates the number of 32-bit items defining the optional shape of this region and therefore also specifies indirectly the total size of this record (12 + N * 4 octets). For compatibility with future version of this RCP PLUS specification, clients SHOULD set this item to 0 for any region which they define or modify, but keep it and the corresponding shape data unmodified for regions they don't change.

hPos/vPos of vertex 1, 2, ..., N

N 32-bit items used for shape data. Each item describes a vertex of a polygonal shape boundary in the order as included in the record (the closing edge between vertex N and vertex 1 is added automatically). The vertices should all be different and the edges between them should have no point in common beside the vertex between them. (If the resulting area inside the polygon is not single connected the behaviour is undefined)

Coordinate System

The rectangles and the future shapes are specified via a virtual coordinate system with a nominal range of 0..32768 x 0..32768 for the usable sensor area (or incoming video). All rectangles and shapes should be completely inside this area, otherwise, the exact behaviour is undefined.

CONF_VIN_BASE_FRAMERATE

Tag code	NumDes	Message	SNMP Support
0x0ad6	cam	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete use CONF_VIDEO_INPUT_FORMAT_EX_VERBOSE
Write	t_dword	access_right_service	not supported

CONF_ILLUMINATION_MODE

Tag code	NumDes	Message	SNMP Support
0x0ae4	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get illumination mode (0: "IR illumination auto"; 1: "IR illumination off"; 2: "IR illumination on")
Write	t_octet	access_right_service	set illumination mode (0: "IR illumination auto"; 1: "IR illumination off"; 2: "IR illumination on")

CONF_MAINS_FREQUENCY

Tag code	NumDes	Message	SNMP Support
0x0af7	-	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the mains frequency. 0=50Hz, 1=60Hz
Write	t_dword	access_right_service	set the mains frequency. 0=50Hz, 1=60Hz

CONF_VIRTUAL_LINES

Tag code	NumDes	Message	SNMP Support
0x0bf4	-	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_minimal	not supported

The command returns which video input lines are virtual and which is the assiciated physical line. A physical line has it's own video signal - that's the usual case. A virtual line is derived from a physical line and supports only a subset of the RCP commands compared to a physical line. Some commands only make sense on physical line (e.g. brightness, contrast, ...). Those fail on virtual lines and must be applied on the associated physcial line instead. The required information can be read out using this command.
Virtual lines are introduced to provide a second view on an existing physical line. This is useful e.g. to have a dewarped image on a 180 or 360 degree camera. Furthermore each virtual line has its own video analysis meta data stream (a.k.a. "VCD stream") which is also derived from the image on the physical line but already preprocessed (cropped, dewarped, ...) to fit the view on the virtual line.

Payload Structure

No. of Entries 2 Byte	**Length per Entry** 2 Byte
First Entry 4 Byte	
... 4 Byte	
Last Entry 4 Byte	

No. of Entries

The number of virtual line descriptions.

Length per Entry

The number of bytes per line description. This value is currently 4 but may be increased later when more information per entry must be provided. New fields will be appended to the entry and should be ignored by the client when unknown.

Entry

No. of virtual Line 2 Byte	**No. of physical Line** 2 Byte

No. of virtual Line

The number of the line which is a virtual line. Counting starts as usual from 1. Every video input line is either a physical line (default) or a virtual line. All lines listed here are virtual lines.

No. of physcial Line

The number of the physcial line which is associated to the named virtual line in the previous field. This line must be addressed for RCP commands that fail on the virtual line.

CONF_VIDEO_LINE_DEWARPING_MODE

Tag code	NumDes	Message	SNMP Support
0x0bf6	line (starting from 1)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Gets the dewarping mode of a virtual line. octet[0] is the selected id. If the request is applied to a physical line, the value 0 is returned. (Do not check the length of the command in the future may additional tags will be added). If only the first Byte is set for addition parameters defaults will be used. See CONF_VIDEO_LINE_DEWARPING_MODE_CAPS for possible values.
Write	p_octet	access_right_minimal	Sets the dewarping mode of a virtual line octet[0] sets the selected id. (Do not check the length of the command in the future may additional tags will be added). If only the first Byte is set for addition parameters defaults will be used. See CONF_VIDEO_LINE_DEWARPING_MODE_CAPS for possible values.

CONF_VIDEO_LINE_DEWARPING_MODE_CAPS

Tag code	NumDes	Message	SNMP Support
0x0bf7	line (starting from 1)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Returns a list of possible dewarping modes of a virtual line. Each mode is a 8 bit value: 0: no dewarping, 1: virtual PTZ, 2: quad view, 3: panoramic, 4: double panoramic, 5: corridor, 6: full panoramic
Write	%	access_right_minimal	%

CONF_VIDEO_LINE_DUO

Tag code	NumDes	Message	SNMP Support
0x0ca9	line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	duo line support (0=off, 1=on), use CONF_DEVICE_CAPABILITIES tag 41 to detect support
Write	t_dword	access_right_service	duo line support (0=off, 1=on), use CONF_DEVICE_CAPABILITIES tag 41 to detect support

CONF_VIDEO_OUT_STANDARD

Tag code	NumDes	Message	SNMP Support
0x0700	output line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	select video output standard: 0=AUTO; 1=PAL; 2=NTSC; 3,4,5,6,7,8,9,10=VGA modes; 11=720p50; 12=720p60; 13=1080i50; 14=1080i60; 15=1080p25; 16=1080p30; 0xff=OFF
Write	t_octet	access_right_service	select video output standard: 0=AUTO; 1=PAL; 2=NTSC; 3,4,5,6,7,8,9,10=VGA modes; 11=720p50; 12=720p60; 13=1080i50; 14=1080i60; 15=1080p25; 16=1080p30; 0xff=OFF

CONF_VIDEO_OUT_STANDARD_FORCE

Tag code	NumDes	Message	SNMP Support
0x070b	output line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	force video output standard (e.g. for HDMI)
Write	t_octet	access_right_service	force video output standard (e.g. for HDMI)

CONF_VIDEO_OUT_CROPPING

Tag code	NumDes	Message	SNMP Support
0x070c	output line	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	enable cropping at video output if the aspect ratio of the incoming video doesn't fit
Write	t_octet	access_right_service	enable cropping at video output if the aspect ratio of the incoming video doesn't fit

CONF_VIDEO_OUT_DISPLAY_FORMAT

Tag code	NumDes	Message	SNMP Support
0x070a	line number	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	(analog) display format: LSB: 0 = AUTO/ANAMORPH; 1 = 4:3; 2 = ANAMORPH 16:9; MSB: flags: 2 = crop, all other reserved (set to 0)
Write	t_word	access_right_service	(analog) display format: LSB: 0 = AUTO/ANAMORPH; 1 = 4:3; 2 = ANAMORPH 16:9; MSB: flags: 2 = crop, all other reserved (set to 0)

CONF_MONITOR_NAME

	Tag code	NumDes	Message	SNMP Support
	0x028a	number of video output line	no	no
	Datatype	**Access Level**	**Description**	
Read	p_unicode	access_right_minimal	read the video monitor name	
Write	p_unicode (max 32 unicode characters)	access_right_service	set the video monitor name	

CONF_VID_DECODER_ON

	Tag code	NumDes	Message	SNMP Support
	0x0264	no	no	no
	Datatype	**Access Level**	**Description**	
Read	flag	access_right_minimal	=1: video decoder is enabled on establishing a new connection; =0 video encoder is not deabled on establishing a new connection	
Write	flag	access_right_service	=1 enables the video decoder on establishing a new connection; =0 disables the video decoder on establishing a new connection	

CONF_LOGO

	Tag code	NumDes	Message	SNMP Support
	0x0939	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	0 := show "logo.h263" from toolkit 1:= no logo	
Write	t_octet	access_right_service	0 := show "logo.h263" from toolkit 1:= no logo 2:= screen saver	

CONF_DEC_SHOW_FREEZE

Tag code	NumDes	Message	SNMP Support
0x092e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	freeze string: 0=off, 1=on, n=on (n * 40ms delayed)
Write	t_octet	access_right_service	freeze string: 0=off, 1=on, n=on (n * 40ms delayed)

CONF_DECODER_MODE

Tag code	NumDes	Message	SNMP Support
0x0812	no	no	no

	Datatype	Access Level	Description
Read	octet	access_right_minimal	decoder mode =0: low delay; =1: A/V sync;
Write	octet	access_right_service	decoder mode =0: low delay; =1: A/V sync;

CONF_DECODER_LAYOUT_LIST

Tag code	NumDes	Message	SNMP Support
0x09a1	video out line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	list of supported layouts; payload is a list of n WORDs. One WORD per supported layout (0001=single view, 0002=quad view, 0004=sixteen, 0077=multi view)
Write	p_octet	access_right_user	not supported

CONF_DECODER_LAYOUT

Tag code	NumDes	Message	SNMP Support
0x09a2	video out line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	first word (2bytes) is the layout, followed by the relative coder list in bytes (000101 for first monitor singleview, 000201020304 for quadview with chronological order)
Write	p_octet	access_right_user	first word is the layout, followed by the relative coder list; extension for multiview transcoder see detailed description

Multiview Payload Structure

Layout 2 Byte	**reserved** 2 Byte
Flags 1 Byte	**YUV** 3 Bytes
EntryList 1 16 Bytes ...	
N x 16 Bytes	**EntryList N** 16 Bytes ...

Layout

Layout (set to 'M' in case of multiview)

Values:

Single	1
Quad	2
Nine	3
Sixteen	4
Multiview	M

reserved

reserved

Flags

bit 0==1 signals RGB instead of YUV

YUV

background YUV

EntryList 1 - N

list of multiviews

16 32

len 1 Byte	**flags** 1 Byte	**xPosition** 1 Byte	**yPosition** 1 Byte
width 1 Byte	**height** 1 Byte	**inst** 1 Byte	**reserved** 1 Byte
hPos 2 Byte		**vPos** 2 Byte	
size 2 Byte		**reserved** 2 Byte	

8 24

len

entry len

flags

Bit 0= flip, bit 1=mirror, bit 2=allFrames

xPosition

x position in grid

yPosition

y position in grid

width

width in grid

height

height in grid

inst

instance

reserved

reserved

hPos

optional ptz hpos (see CONF_ROI)

vPos

optional ptz vpos (see CONF_ROI)

size

optional ptz size(see CONF_ROI)

reserved

reserved

CONF_VIDEO_OUT_STANDARD_SPEC

Tag code	NumDes	Message	SNMP Support
0x0706	yes: corresponding video output standard number (see CONF_VIDEO_OUT_STANDARD)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obtain video output standard specification
Write	p_octet	access_right_service	configure video output standard (use zero payload size to reset to default)

Payload Structure

16 32

pixelClock 4 Bytes						
width 2 Byte		**height** 2 Byte				
border_lft 2 Byte		**border_rgt** 2 Byte				
border_top 2 Byte		**border_bot** 2 Byte				
retrace_width 2 Byte		**retrace_height** 2 Byte				
hsync_start 2 Byte		**vsync_start** 2 Byte				
hsync_width 2 Byte		**vsync_width** 2 Byte				
border_width 2 Byte		**border_height** 2 Byte				
border_red 1 Byte	**border _green** 1 Byte	**border_blue** 1 Byte	**reserved** 4 Bits	**h** 1	**v** 1	**sy** 2
reserved 28 Bytes ...						

8 24

pixelClock

pixel clock frequency in Hz

width

width of active screen in pixel

height

height of active screen in pixel

border_lft

width of the additional border left of the active screen

border_rgt

width of the additional border right of the active screen

border_top

height of the additional border above the active screen

border_bot

height of the additional border below active screen

retrace_width

duration of horizontal blanking period in pixel clock cycles

retrace_height

duration of vertical blanking period in scan line units

hsync_start

start of the horizontal sync impulse in pixel clock cycles past the beginning of the horizontal blanking period (signed 2s-complement)

vsync_start

start of the vertical sync impulse in scan lines past the beginning of the vertical blanking period (signed 2s-complement)

hsync_width

duration of the horizontal sync impulses in pixel clock cycles

vsync_width

duration of the vertical sync impulses in scan line units

border_width

(minimum) border width between the active subscreens in split screen mode

border_height

(minimum) border height between the active subscreens in split screen mode

border_red

intensity of the red channel of the border colour around the active screen and between subscreens

border _green

corresponding intensity of the green channel

border_blue

corresponding intensity of the blue channel

reserved

reserved for future extensions

hsync mode

polarity of horizontal sync signal

Values:

positive polarity (active high)	0
negtive polarity (active low)	1

vsync mode

polarity of vertical sync signal

Values:

positive polarity (active high)	0
negative polarity (active low)	1

sync signals

Values:

separate H-/V-sync signals	0
composite sync signal	1
illegal	2
composite sync on green	3

reserved

reserved for future extensions

Notes

In order to be selectable via CONV_VIDEO_OUT_STANDARD or CONV_VIDEO_OUT_CURRENT_SPEC, the corresponding parameters must comply with the currently active output device limits (see CONF_VIDEO_OUT_MONITOR_SPEC).

The resulting scan line frequency can be calculated by:

f_hsync = pixelClock / (width+border_lft+border_rgt+retrace_width)

The resulting vertical refresh frequency can be calculated by

f_vsync=f_hsync/(height+border_top+border_bot+retrace_height)

Negative values for sync_start and/or vsync_start are not guaranteed to work.

Depending on the hardware, the actual border colour may be different from the selected one and even different between the border around the active screen and the border between the subscreens in split screen mode.

Not all sync mode combinations are supported by the hardware.

The values returned on read may be different from the values set via the last write access, even if the supplied values are within the range supported by the output display as configured via CONF_VIDEO_OUT_MONITOR_SPEC.

CONF_VIDEO_OUT_CURRENT_SPEC

	Tag code	NumDes	Message	SNMP Support
	0x0707	yes: video out line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete
Write	p_octet	access_right_service	obsolete

CONF_VIDEO_OUT_MONITOR_SPEC

Tag code	NumDes	Message	SNMP Support
0x0708	yes: video out line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read active monitor specification of a video out line
Write	p_octet	access_right_service	override default monitor timing spec

Payload Structure

16 32

width_mm 2 Byte		**height_mm** 2 Byte	
native_width 2 Byte		**native_height** 2 Byte	
hSyncMin 4 Bytes			
hSyncMax 4 Bytes			
vSyncMin 4 Bytes			
vSyncMax 4 Bytes			
dotClockMax 4 Bytes			
reserved 23 Bits	**s** 1	**compSync** 4 Bits	**sepSync** 4 Bits
reserved 32 Bytes ...			

8 24

width_mm

physical width of the actual display region in mm

height_mm

physical height of the actual display region in mm

native_width

(maximum) width of the active display in pixel

native_height

(maximum) height of the active display in pixel

hSyncMin

lowest supported scan line (hsync) frequency in Hz

hSyncMax

highest supported scan line (hsync) frequency in Hz

vSyncMin

lowest supported vertical refresh frequency in mHz

vSyncMax

highest supported vertical refresh frequency in mHz

dotClockMax

upper limit on the pixel clock

reserved

reserved for future extensions

syncOnGreen

sync on green support:

Values:

unsupported	0
supported	1

compSync

bitmask of supported composite sync signals:

Values:

h-sync active high, v-sync active high	1
h-sync active low, v-sync active high	2
h-sync active high, v-sync active low	4
h-sync active low, v-sync active low	8

sepSync

bitmask of supported separate sync signals:

Values:

h-sync active high, v-sync active high	1
h-sync active low, v-sync active high	2
h-sync active high, v-sync active low	4
h-sync active low, v-sync active low	8

reserved

reserved for future extensions

Notes

In addition to the specified limits, default modes corresponding to the respective physical connector (e.g. standard VGA timing on a VGA connector, PAL/NTSC on BNC) are typically also assumed to be supported by the display device. The device will only allow video output timings (see CONF_VIDEO_OUT_CURRENT_SPEC, CONF_VIDEO_OUT_STANDARD) within these limits and may use the mentioned default timings as fallback values.

CONF_VIDEO_OUT_OVERSCAN

	Tag code	NumDes	Message	SNMP Support
	0x0709	line number, 0=default line	no	no

	Datatype	Access Level	Description
Read	t_word or t_dword	access_right_minimal	configure overscan area; arg = (monitor_overscan<<16) + intended_overscan; 100 % = 65536, values >= 50 % are forced to tolerable values
Write	t_dword	access_right_service	configure overscan area; arg = (monitor_overscan<<16) + intended_overscan; 100 % = 65536, values >= 50 % are forced to tolerable values

CONF_DECODED_FRAMES

Tag code	NumDes	Message	SNMP Support
0x0a1a	yes (coder number)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of decoded video frames of one deocder instance
Write	%	access_right_service	%

CONF_NBR_OF_VIDEO_OUT

Tag code	NumDes	Message	SNMP Support
0x01d7	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed video outputs
Write	void	access_right_minimal	not supported

CONF_DECODER_DELAY

Tag code	NumDes	Message	SNMP Support
0x0bbe	decoder line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get additional delay for decoder in milliseconds
Write	t_dword	access_right_service	set additional delay for decoder in milliseconds

CONF_DECODING_ERROR

Tag code	NumDes	Message	SNMP Support
0x0bbf	yes (coder number)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get decoding error information for a specific coder in a RCP message
Write	p_octet	access_right_minimal	not supported

Payload Structure

ID 4 Byte
Optional info payload N Bytes

ID

Values:

1 H.264 video resolution exceeded decoder limit.

Payload:
4 Byte "current width"
4 Byte "current height"
4 Byte "max width"
4 Byte "max height"

2 Unknown JPEG format

CONF_STAMP_ATTR_DEC_FREEZE

Tag code	NumDes	Message	SNMP Support
0x0933	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

reserved 2 Byte	**bg_luma** 1 Byte	**bg_chroma** 1 Byte
attributes 4 Bytes		
font_luma 1 Byte	**font_chroma** 1 Byte	**flags** 2 Byte

bg_luma

background color: 8 bit luma (Y)

bg_chroma

background color: 2 x 4 bit chroma (U/V); "lower 4 bit" = U, "upper 4 bit" = V

attributes

Values:

Bit 8	Use custom font and background color given in bg_luma, bg_chroma, font_luma and font_chroma (0=off, 1=on)
Bit 10	Big text background box (0=off, 1=on)

font_luma

font color: 8 bit luma (Y)

font_chroma

font color: 2 x 4 bit chroma (U/V); "lower 4 bit" = U, "upper 4 bit" = V

reserved

Reserved fields must be written as 0

CONF_STAMP_DEC_FREEZE_STRING

Tag code	NumDes	Message	SNMP Support
0x092f	no	yes	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	string displayed on the local monitor if DEC_SHOW_FREEZE is enabled and no vido data is coming in
Write	p_string	access_right_service	string displayed on the local monitor if DEC_SHOW_FREEZE is enabled and no vido data is coming in

CONF_STAMP_ATTR_NO_LINK

Tag code	NumDes	Message	SNMP Support
0x0935	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	12 bytes: byte 0= pos x (0-255), byte 1= pos y (0-255), byte 3-11 reserved (set to zero)
Write	p_octet	access_right_service	12 bytes: byte 0= pos x (0-255), byte 1= pos y (0-255), byte 3-11 reserved (set to zero)

CONF_STAMP_NO_LINK_STRING

Tag code	NumDes	Message	SNMP Support
0x0931	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	string displayed on the local monitor if there is no ethernet link
Write	p_string	access_right_service	string displayed on the local monitor if there is no ethernet link

CONF_ENABLE_OSD

Tag code	NumDes	Message	SNMP Support
0x0810	no	no	no

	Datatype	Access Level	Description
Read	octet	access_right_minimal	osd on trimedia
Write	octet	access_right_service	enable osd on trimedia

CONF_OSD_ACCESS

Tag code	NumDes	Message	SNMP Support
0x098a	0: print to all coders, else, print only to selected absolute coder	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_service	access to the OSD via RCP command

Up to version 4.50, this command only applies to decoders to display text onto the video output. The num field carries the video output number. A value of 0 indicates that the text should be displayed on all screens. Remind that the num value numbering scheme takes quad mode into account for the lines where quad is possible, even if quad mode is not active. (E.g. to have an OSD stamped into the fourth line of a VIP X1600 XFMD, you need to use num value of 10). To delete previously submitted strings, send an empty string (not space characters!).

From version 4.50 on this command applies to encoders as well. In that case, the encoder will embed the text information into the RTP data stream to be overlayed by the receiving decoder. The attribute and flag fields are not applicable to the decoder's video output. The num field applies to the video input number.

Payload Structure

ID 2 Byte		**len** 2 Byte	
x 1 Byte	**y** 1 Byte	**bg_luma** 1 Byte	**bg_chroma** 1 Byte
attributes 4 Bytes			
font_luma 1 Byte	**font_chroma** 1 Byte	**flags** 2 Byte	
String 1 1 Byte	N x 1 Byte	**String N** 1 Byte	

ID

Tag ID, for addressing different strings. Values 0x04-0x08 are possible.

len

Length of the complete payload in bytes. Maximum length is 64 bytes.

x

X-Position (0...255) where the string should be displayed (0 is left)

y

Y-Position (0...255) where the string should be displayed (0 is up)

bg_luma

background color: 8 bit luma (Y)

bg_chroma

background color: 2 x 4 bit chroma (U/V); "lower 4 bit" = U, "upper 4 bit" = V

attributes

Values:

Bit 8	Use custom font and background color given in bg_luma, bg_chroma, font_luma and font_chroma (0=off, 1=on)
Bit 10	Big text background box (0=off, 1=on)
Bit 11	Centered display (0=off, 1=on)
Bit 13-15	Font size (0=small, 1=big)
Bit 20	Draw a box around the text or transparent text background (0=box, 1=transparent)

font_luma

font color: 8 bit luma (Y)

font_chroma

font color: 2 x 4 bit chroma (U/V); "lower 4 bit" = U, "upper 4 bit" = V

flags

Values:

Bit 0	Interpret text as unicode characters (UTF-16)

String 1 - N

The string value which should be displayed

CONF_OSD_POS

Tag code	NumDes	Message	SNMP Support
0x0ae0	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	access to the OSD via RCP command
Write	p_octet	access_right_service	access to the OSD via RCP command

Set or get the position of OSD objects (see CONF_OSD_ACCESS). For "write" the num field carries the video number; 0 indicates that the string positions on all video lines will be set.

WRITE

Payload Structure

ID	x	y
2 Byte	1 Byte	1 Byte

ID

Tag ID, for addressing different strings. Values 0x04-0x08 are possible.

x

X-Position (0...255) where the string should be displayed (0 is left)

y

Y-Position (0...255) where the string should be displayed (0 is up)

READ

Payload Structure (request)

ID
2 Byte

Payload Structure (reply)

ID	x	y
2 Byte	1 Byte	1 Byte

ID

Tag ID, for addressing different strings. Values 0x04-0x08 are possible.

x

X-Position (0...255) where the string is located (0 is left)

y

Y-Position (0...255) where the string is located (0 is up)

CONF_AUDIO_INPUT_LEVEL

Tag code	NumDes	Message	SNMP Support
0x000a	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets current level of audio input. Range: 0 to AUDIO_INPUT_MAX
Write	dword	access_right_service	sets level of audio input. Range: 0 to AUDIO_INPUT_MAX

CONF_AUDIO_OUTPUT_LEVEL

Tag code	NumDes	Message	SNMP Support
0x09b7	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets level of audio output. Range: 0 to AUDIO_OUTPUT_MAX
Write	t_dword	access_right_service	sets level of audio output. Range: 0 to AUDIO_OUTPUT_MAX

CONF_AUDIO_ON_OFF

Tag code	NumDes	Message	SNMP Support
0x000c	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0 = audio mode is not set, 1 = audio mode is set
Write	flag	access_right_service	set audio mode (0=off, 1=on) (has no effect on a running recording, to have the effect in recording, restart of recording is nessessary)

CONF_AUDIO_STARTUP_SOUND

Tag code	NumDes	Message	SNMP Support
0x09b6	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	turn on/off audio startup sound
Write	flag	access_right_service	turn on/off audio startup sound

CONF_AUDIO_NOTIFICATION_SOUND

Tag code	NumDes	Message	SNMP Support
0x0c77	sound type: 1=privacy mode sound, ...	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	read status of audio notification sound (only applicable on certain devices). 0 = do not play sound; 1 = play sound
Write	f_flag	access_right_service	switch audio notification sound on/off (only applicable on certain devices). 0 = do not play sound; 1 = play sound

CONF_AUDIO_INPUT

Tag code	NumDes	Message	SNMP Support
0x09b8	audio line	no	no

	Datatype	Access Level	Description
Read	dword	access_right_minimal	0=Line, 1=Mic
Write	dword	access_right_service	0=Line, 1=Mic, 2=Mute (only supported by arm based products)

CONF_AUDIO_OUTPUT

Tag code	NumDes	Message	SNMP Support
0x09b9	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	0=Decoder, 1=Sine, 2=Loop, 3=Mute, 4=Sample
Write	t_dword	access_right_service	0=Decoder, 1=Sine, 2=Loop, 3=Mute, 4=Sample

CONF_AUDIO_INPUT_MAX

Tag code	NumDes	Message	SNMP Support
0x09ba	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the maximum input level that can be adjusted
Write	%	access_right_service	%

CONF_AUDIO_OUTPUT_MAX

Tag code	NumDes	Message	SNMP Support
0x09bb	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the maximum output level that can be adjusted
Write	%	access_right_service	%

CONF_AUDIO_MIC_LEVEL

Tag code	NumDes	Message	SNMP Support
0x09bc	audio line	no	no

	Datatype	Access Level	Description
Read	dword	access_right_minimal	gets level of mic input. Range: 0 to AUDIO_MIC_MAX
Write	dword	access_right_service	sets level of mic input. Range: 0 to AUDIO_MIC_MAX

CONF_AUDIO_MIC_MAX

Tag code	NumDes	Message	SNMP Support
0x09bd	audio line	no	no

	Datatype	Access Level	Description
Read	dword	access_right_minimal	gets the maximum mic level that can be adjusted
Write	%	access_right_service	%

CONF_AUDIO_LOUDSPEAKER_ON_OFF

Tag code	NumDes	Message	SNMP Support
0x09be	audio line	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	Gets Loudspeaker if on or off. 0=off, 1=on.
Write	flag	access_right_service	Sets Loudspeaker on or off. 0=off, 1=on.

CONF_AUDIO_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x09bf	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets options for Audio, Bit 0=Line In, Bit 1=Line Out, Bit 2=Mic, Bit 3=Loudspeaker
Write	%	access_right_service	%

CONF_AUDIO_INPUT_PEEK

Tag code	NumDes	Message	SNMP Support
0x09c6	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the max iabs of the audio input signal (clears the value)
Write	t_dword	access_right_service	get the max iabs of the audio input signal (clears the value)

CONF_AUDIO_OUTPUT_PEEK

Tag code	NumDes	Message	SNMP Support
0x09c7	audio line	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the max iabs of the audio output signal (clears the value)
Write	t_dword	access_right_service	get the max iabs of the audio output signal (clears the value)

CONF_NBR_OF_AUDIO_OUT

Tag code	NumDes	Message	SNMP Support
0x01d9	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of audio outputs
Write	%	access_right_service	%

CONF_NBR_OF_AUDIO_IN

Tag code	NumDes	Message	SNMP Support
0x01d8	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of audio inputs
Write	%	access_right_service	%

CONF_AUDIO_AAC_BITRATE

Tag code	NumDes	Message	SNMP Support
0x0b9a	audio line (1 or 2)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the bitrate of the AAC encoder (only value of 48000 and 80000 are supported) for the line given by num
Write	t_dword	access_right_service	set the bitrate for the AAC encoder (only value of 48000 and 80000 are supported) on the line given with the num parameter

CONF_VIRTUAL_AUDIO_LINES

Tag code	NumDes	Message	SNMP Support
0x0bf5	-	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_minimal	not supported

The command returns which audio input lines are virtual and which is the associated physical line. A physical line is phyical audio input - that's the usual case. A virtual line is just a copy of a physical line and supports only a subset of the RCP commands compared to a physical line. Some commands only make sense on physical line (e.g. input volume). Those commands fail on virtual lines and must be applied on the associated physcial line instead. The required information can be read out using this command.
Virtual lines are introduced to provide a copy to an existing physical line. A single physical audio input can be associated to multiple video input lines. This keeps our known behaviour to have an audio line with the same number as the video line.

Payload Structure

No. of Entries 2 Byte	**Length per Entry** 2 Byte
First Entry 4 Byte	
... 4 Byte	
Last Entry 4 Byte	

No. of Entries

The number of virtual line descriptions.

Length per Entry

The number of bytes per line description. This value is currently 4 but may be increased later when more information per entry must be provided. New fields will be appended to the entry and should be ignored by the client when unknown.

Entry

No. of virtual Line 2 Byte	**No. of physical Line** 2 Byte

No. of virtual Line

The number of the line which is a virtual line. Counting starts as usual from 1. Every audio input line is either a physical line (default) or a virtual line. All lines listed here are virtual lines.

No. of physcial Line

The number of the physcial line which is associated to the named virtual line in the previous field. This line must be addressed for RCP commands that fail on the virtual line.

CONF_EXTERNAL_CLIENTS_LIST

Tag code	NumDes	Message	SNMP Support
0x0c9d	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	gets a list of installed external clients
Write	p_octet	access_right_service	unsupported

CONF_EXTERNAL_CLIENT

Tag code	NumDes	Message	SNMP Support
0x0c9e	see detailed description	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	two external client inst are available. This commands starts/stops an external client on one inst.

This command lists information about all running external clients or about a specific external client. To get the information of all running clients set the num parameter to 0xFFFF; To get the information of a specific external client the num parameter is set to:

1.) video app: 1. Bytes video line (1...n), 2. Bytes client inst (0,1)

2.) user app: 1. null, 2. Bytes client inst (0-16)

The response payload consist of a sequence of

16

line	inst	length	name
1 Byte	1 Byte	1 Byte	length Bytes

8

CONF_EXTERNAL_CLIENT_EVENT

Tag code	NumDes	Message	SNMP Support
0x0cdd	-	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_internal	system-internal callback (passed on by CExtensionManager) to handle some extension event
Write	p_octet	access_right_internal	system-internal callback (passed on by CExtensionManager) to handle some extension event

CONF_ALARM_INPUT_LH_VAL

Tag code	NumDes	Message	SNMP Support
0x008d	alarm input	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	1=alarm pin high active; 2=alarm pin low active
Write	t_octet	access_right_service	1=set alarm pin high active; 2=set alarm pin low active

CONF_INPUT_PIN_NAME

Tag code	NumDes	Message	SNMP Support
0x0108	alarm in	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read the description for an input alarm
Write	p_unicode (max 32 unicode characters)	access_right_service	set the description for an input alarm

CONF_ALARM_INPUT_SUPERVISED

	Tag code	NumDes	Message	SNMP Support
	0x0b87	alarm input (beginning with '1')	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	1= alarm pin is set to supervised mode; 0= alarm pin is set to normal mode (non-supervised)	
Write	t_octet	access_right_service	1=set alarm pin to supervised mode; 0=set alarm pin to normal mode (non-supervised)	

CONF_ALARM_INPUT_CAPABILITIES

	Tag code	NumDes	Message	SNMP Support
	0x0c6a	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	read capabilities of the available alarm inputs; 1st DWORD: number of available alarm inputs; 2nd DWORD: capabilities of the first alarm input; 3rd DWORD: capabilities of the second alarm input; ... Each capability is signalled as one bit within the corresponding DWORD: bit 0 = supervised mode possible, bit 1-31 = reserved	
Write	p_octet	access_right_service	not supported	

CONF_EVENT_SCRIPT

	Tag code	NumDes	Message	SNMP Support
	0x09f3	no	yes (returns script syntax error when not successful)	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_live	get the compressed Alarm Task Script	
Write	p_octet	access_right_service	write a new Alarm Task Script; note: script must be compressed (zipped)	

CONF_VCD_OPERATOR_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0a1b	video line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read parametrization of vcd operator which corresponds to line
Write	p_octet	access_right_service	configures the vcd manager. if configuration fails an error is returned.

CONF_VIPROC_RE_TASK_NAMES

Tag code	NumDes	Message	SNMP Support
0x0b2b	video line	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the current rule engine task names which corresponds to line
Write	p_octet	access_right_service	not supported, read only

This command reads out the task names of the current configured rule engine.

Response Structure

Algorithm type 2 Byte		Number of tasks 2 Byte
Task id 1 Byte	**Task type** 1 Byte	**Name** 64 Byte
...		
Task id 1 Byte	**Task type** 1 Byte	**Task name** 64 Byte

Algorithm type

1 IVA algorithm

2 IVA flow algorithm

4 Motion algorithm

Number of tasks

Number of configured user tasks.

Task id

Id of the user task. Defines the alarm bit of the CONF_VIPROC_ALARM message.

Task type

1 Object in field

2 Line crossing

3 Loitering

5 ConditionChange

6 FollowingRoute

7 Tampering

8 RemovedObject

9 IdleObject

10 EnteringField

11 LeavingField

11 LeavingField

13 SimilaritySearch

14 CrowdDetection

15 Counter

16 BEV people counter

17 Line Counter

18 Occupancy

19 Man over Board

40 FlowInField

41 CounterFlowInField

42 MotionInField

Task name

The task name is UTF8 encoded and maximal 64 byte.

CONF_VIPROC_DETECTOR_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0c96	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	reads VCA detector params
Write	p_octet	access_right_service	sets VCA detector params

CONF_RAW_IMAGE

Tag code	NumDes	Message	SNMP Support
0x0c98	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	not supported

In Payload Structure

16	32
Requested format 2 Byte	**Selected Scaler** 2 Byte
8	24

Available Scalers

0x0001	VCA Scaler (**default**)

Available formats

0x0000	Raw YUV (**default**)
0x0001	TIFF encoded raw

The Raw YUV format begins with the following header:

Image width 4 Byte
Image height 4 Byte
YUV format 4 Byte

The YUV format is encoded as an integer with the following meaning...

Available formats

1	420 planar, YYYY...UUUU...VVVV
-1	420 chroma interleaved, YYYY....UVUVUV
2	422 planar, YYYY...UUUU...VVVV
-2	422 chroma interleaved, YYYY....UVUVUV

CONF_VCA_INPUT_FORMAT

Tag code	NumDes	Message	SNMP Support
0x0ca7	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
Image width 4 Byte	
Image height 4 Byte	
8	24

CONF_ALARM_CONNECT_TO_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x0081	destination IP number	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	read the alarm IP using string notation (xxx.xxx.xxx.xxx)	
Write	p_string	access_right_service	set alarm IP using string notation (xxx.xxx.xxx.xxx)	

CONF_ALARM_CONNECT_TO_IP

	Tag code	NumDes	Message	SNMP Support
	0x0041	destination IP number	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	read the connect on alarm event IP address	
Write	t_dword	access_right_service	specify the connect on alarm event IP address	

CONF_NBR_OF_ALTERNATIVE_ALARM_IPS

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

CONF_AUTO_DISCONNECT_TIME

Tag code	NumDes	Message	SNMP Support
0x030d	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	0=no auto disconnect; 0
Write	t_dword	access_right_service	0=no auto disconnect; 0

CONF_DEFAULT_CONNECTION_MODE

Tag code	NumDes	Message	SNMP Support
0x0289	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get the default connection mode: 0=MPEG2, 1=MPEG4, 2=MPEG4/AVC, 3=JPEG, 4=ANY
Write	t_octet	access_right_service	sets the default connection mode: 0=MPEG2, 1=MPEG4, 2=MPEG4/AVC, 3=JPEG, 4=ANY

CONF_STD_MEDIA_CONNECTION_DIRECTION

Tag code	NumDes	Message	SNMP Support
0x030c	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	=0: outgoing media for alarm connections; =1: req. incoming media for alarm connections; =2: req. bidirectional media for alarm connections
Write	t_octet	access_right_service	=0: outgoing media for alarm connections; =1: req. incoming media for alarm connections; =2: req. bidirectional media for alarm connections

CONF_STD_MEDIA_ENCAPSULATION_PROTOKOL

Tag code	NumDes	Message	SNMP Support
0x0309	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	defines the standard media encapsulation protokol: 1=RTP over UDP, 2=TCP
Write	t_octet	access_right_service	defines the standard media encapsulation protokol: 1=RTP over UDP, 2=TCP

CONF_DEFAULT_CAM

Tag code	NumDes	Message	SNMP Support
0x01af	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	reads the default camera for alarm connections
Write	t_octet	access_right_service	defines the default camera for alarm connections

CONF_ALARM_CONNECTION_DESTINATION_PORT

Tag code	NumDes	Message	SNMP Support
0x0a15	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	returns the network destination port on outgoing alarm connections
Write	t_word	access_right_service	selects the destination network port for outgoing alarm connections; all other than 1756 (default RCP port) will use the HTTP tunneling protocol. In this case, this must be the HTTP/HTTPS port of the called host.

CONF_ALARM_CONNECTION_USE_SSL

Tag code	NumDes	Message	SNMP Support
0x0a16	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	use RCP SSL for outgoing alarm connections
Write	flag	access_right_service	use RCP SSL for outgoing alarm connections

CONF_CONNECT_URL

Tag code	NumDes	Message	SNMP Support
0x0a1e	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the connect url
Write	p_string	access_right_service	set the connect url

CONF_VIDEO_ALARM_STATE

Tag code	NumDes	Message	SNMP Support
0x01c2	video line	yes	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=video alarm is off; 1=video alarm is on (no video signal detected at video input)
Write	void	access_right_minimal	not supported

CONF_INPUT_PIN_STATE

Tag code	NumDes	Message	SNMP Support
0x01c0	alarm in	yes	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=alarm input off; 1= alarm input on
Write	void	access_right_minimal	not supported

CONF_MOTION_ALARM_STATE

Tag code	NumDes	Message	SNMP Support
0x01c3	video line	yes	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=motion alarm off; 1=motion alarm on (at elast one of the CONF_VIPROC_ALARM alarmbit (except the video loss alarm) is set
Write	flag	access_right_minimal	not supported

CONF_RELAY_OUTPUT_STATE

Tag code	NumDes	Message	SNMP Support
0x01c1	relay output	yes	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	returns the current logical level of an relay output
Write	flag	access_right_user	set the logical level of an relay output

CONF_RELAY_OUTPUT_MODE

Tag code	NumDes	Message	SNMP Support
0x0bdb	output number	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	1: output pin is "normally open"; 2: output pin is "normally closed"; (Notice: the relay output mode can also be set via the Alarm Task Script Language)
Write	t_octet	access_right_service	1: set output pin to "normally open"; 2: set output pin to "normally closed"; (Notice: the relay output mode can also be set via the Alarm Task Script Language; should not be used in parallel)

CONF_NBR_OF_ALARM_IN

Tag code	NumDes	Message	SNMP Support
0x01db	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed alarm input contacts (possible masterswitch included)
Write	void	access_right_internal	not supported

CONF_NBR_OF_ALARM_OUT

Tag code	NumDes	Message	SNMP Support
0x01dc	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed alarm output contacts
Write	void	access_right_internal	not supported

CONF_NBR_OF_MOTION_DETECTORS

Tag code	NumDes	Message	SNMP Support
0x09af	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed motion detectors
Write	%	access_right_internal	%

CONF_HD_MGR_SIGNAL_ALARM

Tag code	NumDes	Message	SNMP Support
0x0915	no	no	no

	Datatype	Access Level	Description
Read	-	access_right_minimal	not supported
Write	flag	access_right_service	signal alarm: only one alarm per second is allowed,

CONF_VIRTUAL_ALARM_STATE

Tag code	NumDes	Message	SNMP Support
0x0a8b	virtual alarm number	yes	yes

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=virtual alarm off; 1=virtual alarm on
Write	flag	access_right_user	0=virtual alarm off; 1=virtual alarm on

CONF_SET_VIRTUAL_ALARM_ID

Tag code	NumDes	Message	SNMP Support
0x0b41	virtual alarm number	no	no

	Datatype	Access Level	Description
Read	p_unicode (max 32 unicode characters)	access_right_minimal	not supported
Write	p_unicode (max 32 unicode characters)	access_right_service	set virtual alarm with num ## to active and puts unicode string in all meta data stream as an alarm text event (will be recorded in all active recordings). At maximum 31 unicode characters are accepted (32th character has to be the termination character), longer input will be truncated.

CONF_NBR_OF_VIRTUAL_ALARMS

Tag code	NumDes	Message	SNMP Support
0x0aed	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	number of virtual alarm inputs
Write	t_dword	access_right_internal	not supported

CONF_MANIPULATION_ALARM_STATE

Tag code	NumDes	Message	SNMP Support
0x0af0	manipulation alarm nbr	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=manipulation alarm off; 1=manipulation alarm on (notice: for supervised alarm inputs the according CONF_INPUT_PIN_STATE also indicates an alarm in this case)
Write	flag	access_right_internal	not supported

CONF_NBR_OF_MANIPULATION_ALARMS

Tag code	NumDes	Message	SNMP Support
0x0af1	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	number of manipulation alarms
Write	t_dword	access_right_internal	not supported

CONF_VIRTUAL_ALARM_PARAMETER

	Tag code	NumDes	Message	SNMP Support
	0x0ba6	yes (virtual alarm nbr 1 ... n)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	read not supported, see detailed description	
Write	p_octet	access_right_service	activate/deactivate virtual alarms with parameter, see detailed description	

Payload Structure

flags 8 Bits	**reserved** 3 Bytes
alarm content handle 32 Bits	
pre alarm time 32 Bits	
post alarm time 32 Bits	
user data len 32 Bits	
user data 0 to 4000 Bytes ...	
...	

flags

Flags:

ALARM_ACTIVATE	0x01	activate the alarm
ALARM_DEACTIVATE	0x02	deactivate the alarm
ALARM_SINGLE_SHOT	0x03	activate and then deactivate the alarm (ALARM_ACTIVATE | ALARM_DEACTIVATE)
ALARM_FILE_PROTECTION	0x04	protects the file, which contains the alarm video data (only on activate or shot)
ALARM_IGNORE_ALARM_TIME_PARAMETER	0x08	the pre and post alarm time has no meaning, parameter from the record profiles will be used instead
ALARM_USER_DATA_1X	0x10	user data will be recorded only once

alarm content handle

handle for the alarm(required for deactivation or retrigger, returned in reply on activation) non zero value or zero for wild card in case of deactivate

pre alarm time

pre alarm time in seconds

post alarm time

post alarm time in seconds

user data len

length in bytes of the following user data, max 4000 bytes

user data

user data which will be recorded repeatedly, as long as the alarm is aktive or while within the post alarm time of the alarm

VIRTUAL_ALARM_PARAMETER

This command is used to trigger a virtual alarm along with parameters for alarm recording. The alarm can be activated, deactivated or triggered as single shot, which is an activate and deactivate within on command. Its also possible to trigger a silent alarm, by setting the activate and deactivate flag to zero. This works like a single shot alarm, the user data will be recorded and backups between the recordings will be scheduled, but the virtual alarm won't be triggered. If silent alarm is used, the num parameter doesn't choose the virtual alarm number, but it is mapped directly to the recording line. The command can have user data which can be recorded as meta data along to the video data. Also the pre and post alarm time can be specified but is only valid, if the alarm recording is done via backuped recording setup and the Flag ALARM_IGNORE_ALARM_TIME_PARAMETER isn't set. This will cause an alarm backup from the recording (primary or secondary, depending on the virtual alarm configuration and encoder index configuration) with this parameter, but the pre alarm time is limited by the max pre alarm time which is the same as the configured prealarm time of the secondary recording on the internal storage. It is also limited by the storage space of the internal storage. The max

pre alarm time is also the delay for the recording. That means the recording of the alarm file triggered by an alarm will be started after this delay expires. Until that time the records of that file are only available on the internal storage. When a new alarm is triggered by activate, the reply returns an alarm content handle (only on activation, not on single shot). The limit for open content handle at the same time per line (includes also deactivated alarms or alarm shots within the post alarm time) is 32. This handle can be used to deactivate the alarm later by sending this command with the handle in the payload or it can be used to retrigger the alarm with new user data. For retrigger the playload looks the same as for activate, except a valid handle is required. Only for deactivation or retrigger of the alarm, the handle is required for in payload. A valid handle is a non zero handle, if the handle is set to zero, it will be interpreted as wild card on deactivate, which will deactivate/close all active handles or as invalid. In case of activate, a new handle will be opened and returned in the reply, if the handle is invalid/zero. If a handle(or all handles) will be deactivated/closed, it will decay with the post alarm time. An alarm can also used on first activation to add the "protected" attribute to the file, which includes the record data of the alarm. The limit for pending intervals for protection are limited to 32 per line. Pending is an interval in the time from the occurence of the protecting alarm and the backup of the alarm record data.

CONF_PIR_SENSITIVITY

Tag code	NumDes	Message	SNMP Support
0x0bbb	ignored	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	Sensitivity of PIR sensor: 0=disabled, 1=lowest..10=highest
Write	t_octet	access_right_service	Sensitivity of PIR sensor: 0 = PIR disabled, 1 = lowest .. 10 = highest

CONF_PIR_STATUS

Tag code	NumDes	Message	SNMP Support
0x0bbc	ignored	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_user	obsolete - Raw ADC value of the PIR sensor.
Write	t_dword	access_right_service	not supported

CONF_NBR_OF_PIR

Tag code	NumDes	Message	SNMP Support
0x0c1a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of installed PIR sensors
Write	t_dword	access_right_internal	not supported

CONF_PIR_ALARM_STATE

Tag code	NumDes	Message	SNMP Support
0x0c1b	PIR sensor nbr	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	0=PIR alarm off; 1=PIR alarm on
Write	flag	access_right_internal	not supported

CONF_ILLUMINATOR_OPTIONS

Tag code	NumDes	Message	SNMP Support
0x0c1c	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns the number of installed white-light illuminators and their capabilities; 1st WORD: number of illuminators; 2nd WORD: capabilities of first illuminator, 3rd WORD: capabilities of second illuminator..., etc.; Available capabilities: bit0=dimmable (by CONF_ILLUMINATION_INTENSITY)
Write	p_octet	access_right_internal	not supported

CONF_ILLUMINATOR_STATE

Tag code	NumDes	Message	SNMP Support
0x0c1d	illuminator nbr	yes	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get state of white-light illumination; 0=illuminator off; 1=illuminator on
Write	t_dword	access_right_user	switch white-light illumination; 0=illuminator off; 1=illuminator on; (notice: to change the illumination intensity, see command CONF_ILLUMINATION_INTENSITY)

CONF_ILLUMINATION_INTENSITY

Tag code	NumDes	Message	SNMP Support
0x0c22	illuminator nbr	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the white-light illumination intensity; range is from 1-100; 1->darkest; 100->brightest
Write	t_word	access_right_user	set the white-light illumination intensity; range is from 1-100; 1->darkest; 100->brightest; (notice: to switch the illuminator on/off, use command CONF_ILLUMINATOR_STATE)

CONF_VCA_BRIGHTNESS_INFO

Tag code	NumDes	Message	SNMP Support
0x0c25	video line	yes	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	luminance (1..255), the value 0 means N/A
Write	p_octet	access_right_user	not supported

CONF_VCA_SUGGESTED_FIREMASK

Tag code	NumDes	Message	SNMP Support
0x0c31	video line	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read suggested fire mask from algorithm (payload beyond this document)
Write	t_octet	access_right_user	not supported

CONF_RELAIS_NAME

Tag code	NumDes	Message	SNMP Support
0x0109	relay output	no	no

	Datatype	Access Level	Description
Read	p_unicode	access_right_minimal	read the description for a relay output
Write	p_unicode (max 32 unicode characters)	access_right_service	set the description for a relay output

CONF_RELAIS_SWITCH

Tag code	NumDes	Message	SNMP Support
0x0094	relay output	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	return the current relay state
Write	flag	access_right_user	toggle the logical level of a relay output (0->1; 1->0) (notice: a flag payload value is not necessary, i.e. ignored)

CONF_SERIAL_PORT_APP_VAL

Tag code	NumDes	Message	SNMP Support
0x01f1	physical port	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	0=serial port application is in terminal mode; 0xFF=serial port application is in transparent datamode; values 1-10: camera type mode; 0xF8=serial port application is in a mode where incoming conf_bicom commands are piped to the serial port, 0xF8 is only applicable on encoder devices and only for video line 1, for 0xF8 a device reboot is necessary!
Write	t_octet	access_right_service	0=sets the serial port application in terminal mode; 0xFF=sets the serial port application in transparent datamode; values 1-10: sets to camera type mode; 0xF8=sets the serial port application into a mode where incoming conf_bicom commands are piped to the serial port, 0xF8 is only applicable on encoder devices and only for video line 1, for 0xF8 a device reboot is necessary!

CONF_SERIAL_PORT_RATE

Tag code	NumDes	Message	SNMP Support
0x027e	physical port	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	bitrate=300; bitrate=600; bitrate=1200; bitrate=2400; bitrate=4800; bitrate=9600; bitrate=19200
Write	t_dword	access_right_service	bitrate=300; bitrate=600; bitrate=1200; bitrate=2400; bitrate=4800; bitrate=9600; bitrate=19200

CONF_SERIAL_PORT_BITS

Tag code	NumDes	Message	SNMP Support
0x027f	physical port	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	7=seven data bits; 8=eight data bits
Write	t_octet	access_right_service	7=set to seven data bits; 8=set to eight data bits

CONF_SERIAL_PORT_STBITS

Tag code	NumDes	Message	SNMP Support
0x0280	physical port	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	1=modemport stopbits are set to 1; 2=modemport stopbits are set to 2
Write	t_octet	access_right_service	1=modemport stopbits are set to 1; 2=modemport stopbits are set to 2

CONF_SERIAL_PORT_PAR

Tag code	NumDes	Message	SNMP Support
0x0281	physical port	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	0=no parity; 1=even parity; 2=odd parity
Write	t_octet	access_right_service	0=set to no parity; 1=set to even parity; 2=set to odd parity

CONF_SERIAL_PORT_MODE_VAL

Tag code	NumDes	Message	SNMP Support
0x0208	physical port	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	1=port mode is set to RS232; 2=port mode is set to RS422/RS485 (NOT avalibale in conventional VideoJet)
Write	t_octet	access_right_service	1=set port mode to RS232; 2=set port mode to RS422/RS485

CONF_SERIAL_PORT_HD_MODE_VAL

Tag code	NumDes	Message	SNMP Support
0x020b	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	=0: halfduplex mode is set to off; =1:halfduplex mode is set to on; =2: halfduplex mode is set to on and buffered data (NOT avalibale in conventional VideoJet)
Write	t_octet	access_right_service	=0: set halfduplex mode to off; =1:set halfduplex mode to on; =2: set halfduplex mode to on and buffered data

CONF_SERIAL_PORT_HANDSHAKE

Tag code	NumDes	Message	SNMP Support
0x09a9	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get type of handshake (internal use only)
Write	t_octet	access_right_service	set type of handshake (internal use only)

CONF_SERIAL_ACTS_ACCESS_RIGHTS

Tag code	NumDes	Message	SNMP Support
0x0b1a	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	returns access rights of ACTS/OSRD commands which are sent via transparent data. 0: user rights (mainly PTZ) (default), 1: service rights (change settings)
Write	t_octet	access_right_service	set access rights of ACTS/OSRD commands which are sent via transparent data. 0: user rights (mainly PTZ) (default), 1: service rights (change settings)

CONF_BICOM_COMMAND

Tag code	NumDes	Message	SNMP Support
0x09a5	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_user	Sends a BICOM read or write command to the local camera frontend, see detailed description . Notice: for some BICOM commands an access level higher than 'l_user' is needed. For details about this see Appendix 'Bicom Command Access Levels' of this document.

Bicom message over RCP command

Payload Structure

Without Lease Time

8	24	40	48
Flags 1 Byte	**Bicom Server ID** 2 Byte	**Object ID** 2 Byte	**Operation** 1 Byte
Bicom Payload n Bytes			

With Lease Time

8	24		56
Flags 1 Byte	**Lease time** 2 Bytes	**Lease time id** 4 Bytes	
Bicom Server ID 2 Byte	**Object ID** 2 Byte	**Operation** 1 Byte	**Bicom Payload** n Bytes

Flags

Transmission-Flags

Values:

Return_Payload (Must be set to 1 if return payload is expected)	Bit 0
Best_Effort (Set to 1 to transmitt as best effort frame)	Bit 1
Native_Errors (Set to 1 to receive the native BICOM errors)	Bit 2
Lease_Time_Available (Set to 1 if a lease time is included in the request)	Bit 3
unused set to 0	Bit 4
unused set to 0	Bit 5
unused set to 0	Bit 6
Flags_Available (Must be always set to 1)	Bit 7

Lease time

Time period in seconds the access should be blocked for other clients.

NOTE: this field has to be signaled in little endian mode.

Lease time id

Random number generated by the client. If a lease time > 0 is provided with the first access, further accesses during the lease time are only possible if the same lease time id is provided.

Bicom Server ID

Server ID, e.g. 0x0002 for "Device Server" (See BICOM application documentation)

Object ID

Object ID, e.g. 0x0100 for "Type" (See BICOM application documentation)

Operation

See BICOM application documentation

Values:

GET	0x01

SET	0x02
SET_GET	0x03
INC	0x04
INC_GET	0x05
DEC	0x06
DEC_GET	0x07
SET_DFLT	0x08
SET_GET_DFLT	0x09

Bicom Payload

Bicom Payload

Example

RCP Request

81	Return payload expected
00 02	Server ID = 2 ("Device server")
01 50	Object "IdString"
01	Operation GET

RCP Reply

81	Return payload expected
00 02	Server ID = 2 ("Device server")
01 50	Object "IdString"
01	Operation GET
00 68 00 68 00 68 00 68 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20	34 Bytes Unicode ID String

Bicom event over RCP message

Payload Structure

16	32	40
Bicom Server ID 2 Byte	**Object ID** 2 Byte	**Operation** 1 Byte
Bicom Payload n Bytes		

Bicom Server ID

Server ID, e.g. 0x0004 for "Camera Server" (See BICOM application documentation)

Object ID

Object ID, e.g. 0x0190 for "Colour" (See BICOM application documentation)

Operation

See BICOM application documentation

Values:

EVENT	0x70 - 0x7F

Bicom Payload

Bicom Payload

Example

RCP message

00 04	Server ID = 4 ("Camera server")
01 90	Object "Colour"
70	Operation EVENT
00 00	2 Byte unsigned short: Colour mode is "B/W"

CONF_BICOM_SRV_CONNECTED

Tag code	NumDes	Message	SNMP Support
0x0a19	no	yes	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	return true if the device is connected to the BICOM server (Camera)
Write	%	access_right_user	%

CONF_BICOM_SUBCOMPONENTS_LIST

Tag code	NumDes	Message	SNMP Support
0x0aa5	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Read List of FamilyIDs and Firmware Versions for Subcomponents of Gen4 (Future Dinion?)
Write	p_octet	access_right_internal	Read List of FamilyIDs and Firmware Versions for Subcomponents of Gen4 (Future Dinion?)

CONF_KBD_CONFIG_CAMERA

	Tag code	NumDes	Message	SNMP Support
	0x0a31	yes (camera number on keyboard)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read configuration of camera numbers on keyboard
Write	p_octet	access_right_service	configure camera numbers on keyboard; assign parameters to a number (given by num)

Payload Structure

16 32

IP 4 Bytes			
Line 1 Byte	**Coder** 1 Byte	**Preset** 1 Byte	**Reserved** 1 Byte

8 24

IP

Encoder/Camera IP

Line

Video input line

Coder

Relative coder number (relative to line)

Preset (optional)

(Dome-) Preset Position. (Every Preset of a Dome can be treated as a seperate camera).

CONF_KBD_CONFIG_CAMERA_STR

Tag code	NumDes	Message	SNMP Support
0x0ba3	yes (camera number on keyboard)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read configuration of camera numbers on keyboard
Write	p_octet	access_right_service	configure camera numbers on keyboard; assign parameters to a number (given by num)

Payload Structure

16 32

Line 1 Byte	**Coder** 1 Byte	**Preset** 1 Byte	**URL Length** 1 Byte
URL N Bytes			

8 24

URL

Encoder/Camera URL

Line

Video input line

Coder

Relative coder number (relative to line)

Preset (optional)

(Dome-) Preset Position. (Every Preset of a Dome can be treated as a seperate camera).

URL Length

Length of the following url (actual limit is 80).

CONF_KBD_CONFIG_MONITOR

	Tag code	NumDes	Message	SNMP Support
	0x0a32	yes (monitor number on keyboard)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read configuration of monitor numbers on keyboard
Write	p_octet	access_right_service	configure monitor numbers on keyboard; assign parameters to a number (given by num)

Payload Structure

16 32

IP 4 Bytes			
Line 1 Byte	**Coder** 1 Byte	**Reserved** 1 Byte	**Reserved** 1 Byte

8 24

IP

Decoder/Monitor IP

Line

Video output line

Coder

Relative coder number (relative to line)

CONF_KBD_CONFIG_MONITOR_STR

	Tag code	NumDes	Message	SNMP Support
	0x0ba4	yes (monitor number on keyboard)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read configuration of monitor numbers on keyboard
Write	p_octet	access_right_service	configure monitor numbers on keyboard; assign parameters to a number (given by num)

Payload Structure

16 32

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

8 24

URL

Decoder/Monitor URL

Line

Video output line

Coder

Relative coder number (relative to line)

URL Length

Length of the following url (actual limit is 80).

CONF_KBD_CONNECT_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0a33	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read parameters for connections established via keyboard
Write	p_octet	access_right_service	set parameters for connections established via keyboard

Payload Structure

16	32
Flags 2 Bytes	**Reserved** 2 Bytes
8	24

Flags (optional)

Values:

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

CONF_KBD_PASSWORD_CAMERA

Tag code	NumDes	Message	SNMP Support
0x0a34	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	get the password (scrambled) for all cameras used on keyboard
Write	p_string	access_right_service	deposit a password for all cameras used on keyboard

CONF_KBD_PASSWORD

Tag code	NumDes	Message	SNMP Support
0x0a69	no	no	no

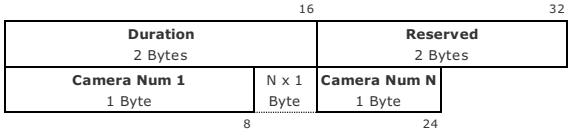
	Datatype	Access Level	Description
Read	p_string	access_right_service	get the password (scrambled) of the keyboard
Write	p_string	access_right_service	deposit a password for the keyboard

CONF_KBD_CONFIG_SALVO

Tag code	NumDes	Message	SNMP Support
0x0a3e	yes (salvo number)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read configuration of salvos for keyboard
Write	p_octet	access_right_service	configure salvos used for keyboard; assign parameters to a salvo number (given by num)

Payload Structure



Duration

Duration of one salvo position in seconds

Camera Num N

Camera number for salvo position N
(note: Camera number has to be specified using CONF_KBD_CONFIG_CAMERA)

CONF_KBD_KEY_LABEL

	Tag code	NumDes	Message	SNMP Support
	0x0a44	yes (key number; must be >0)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	read a label of a key used for the alarm task editor. The label is shown on the keyboard display.	
Write	p_string	access_right_service	deposit a label for a key used for the alarm task editor. The label is shown on the keyboard display.	

CONF_KBD_SET_ALARM

	Tag code	NumDes	Message	SNMP Support
	0x0a68	no	no	no
	Datatype	**Access Level**	**Description**	
Read	-	access_right_minimal	not supported	
Write	f_flag	access_right_minimal	send alarm to keyboard	

CONF_TPPP_PORT

	Tag code	NumDes	Message	SNMP Support
	0x0ca8	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_word	access_right_minimal	gets the port of the third party protocol proxy (0: off)	
Write	t_word	access_right_service	sets the port of the third party protocol proxy (0: off)	

CONF_MAC_ADDRESS

Tag code	NumDes	Message	SNMP Support
0x00bc	yes	no	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read out the systems MAC address numdes=1 EthernetMAC, numdes=2 WlanMAC, numdes=3 BluetoothMAC
Write	void	access_right_minimal	not supported

CONF_IP

Tag code	NumDes	Message	SNMP Support
0x0001	0,1 primary ip, 2 auto ip	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the unit's IP address
Write	t_dword	access_right_service	set the unit's IP address

CONF_IP_STR

Tag code	NumDes	Message	SNMP Support
0x007c	0,1 primary ip, 2 auto ip	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the unit's IP address using string notation (xxx.xxx.xxx.xxx)
Write	p_string	access_right_service	set unit's IP address using string notation (xxx.xxx.xxx.xxx)

CONF_SUBNET

	Tag code	NumDes	Message	SNMP Support
	0x0002	0,1 primary ip subnet, 2 auto ip subnet	no	yes
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	read the unit's subnet	
Write	t_dword	access_right_service	set the unit's subnet	

CONF_SUBNET_STR

	Tag code	NumDes	Message	SNMP Support
	0x007d	0,1 primary ip subnet, 2 auto ip subnet	no	yes
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	read the subnet mask using string notation (xxx.xxx.xxx.xxx)	
Write	p_string	access_right_service	set subnet mask using string notation (xxx.xxx.xxx.xxx)	

CONF_GATEWAY_IP_V6_STRING

	Tag code	NumDes	Message	SNMP Support
	0x0b11	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	IPv6 Gateway IP string or domain name	
Write	p_string	access_right_service	IPv6 Gateway IP string or domain name	

CONF_IP_V6_PREFIX_LEN

Tag code	NumDes	Message	SNMP Support
0x0b05	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get unit's IPv6 address prefix length (Manually assigned IP)
Write	t_octet	access_right_service	get unit's IPv6 address prefix length (Manually assigned IP)

CONF_IP_V6_STR

Tag code	NumDes	Message	SNMP Support
0x0b06	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	Manually assigned IPv6 String or domain name
Write	p_string	access_right_service	Manually assigned IPv6 String or domain name

CONF_STATELESS_IP_V6_PREFIX_LEN

Tag code	NumDes	Message	SNMP Support
0x0bc6	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get unit's IPv6 address prefix length (Automatic assigned IP)
Write	t_octet	access_right_service	get unit's IPv6 address prefix length (Automatic assigned IP)

CONF_STATELESS_IP_V6_STR

Tag code	NumDes	Message	SNMP Support
0x0bc7	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	IPv6 String (Automatic assigned IP)
Write	p_string	access_right_service	IPv6 String (Automatic assigned IP)

CONF_GATEWAY_IP_STR

Tag code	NumDes	Message	SNMP Support
0x007f	no	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the gateway IP using string notation (xxx.xxx.xxx.xxx)
Write	p_string	access_right_service	set gateway IP using string notation (xxx.xxx.xxx.xxx)

CONF_DNS_SERVER_IP

Tag code	NumDes	Message	SNMP Support
0x0a1f	yes	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get dns server ip/ipv6, num = 0 or 1 sets primary dns, num = 2 sets secondary dns
Write	t_dword	access_right_service	set dns server ip/ipv6, num = 0 or 1 sets primary dns, num = 2 sets secondary dns

CONF_DNS_SERVER_IP_STRING

Tag code	NumDes	Message	SNMP Support
0x0b49	yes	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get dns server ip/ipv6, num = 0 or 1 sets primary dns, num = 2 sets secondary dns
Write	p_string	access_right_service	set dns server ip/ipv6, num = 0 or 1 sets primary dns, num = 2 sets secondary dns

CONF_APPLY_NETWORK_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x0cbe	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_service	not supported
Write	t_dword	access_right_service	apply the configured network settings at runtime

CONF_ETH_LINK

Tag code	NumDes	Message	SNMP Support
0x092d	yes; num=0 is the mode of the external state on single ethernet port units or the internal link mode on multiple ethernet port units; num>=1 is the mode of the corresponding port on multiple ethenet port units (SFP Fiber port is always the last counted port). On module based units, only the master module is capable of setting the external port modes.	no	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	(half duplex=HD, full duplex =FD) 0=auto, 1=10MbitHD, 2=10MbitFD, 3=100MbitHD, 4=100MbitFD
Write	t_octet	access_right_service	(half duplex=HD, full duplex =FD) 0=auto, 1=10MbitHD, 2=10MbitFD, 3=100MbitHD, 4=100MbitFD, 5=1000MbitFD, 8=10GbitFD

CONF_ETH_LINK_STATUS

Tag code	NumDes	Message	SNMP Support
0x0a24	yes; num=0 returns the status of the external state on single ethernet port units or the internal link state on multiple ethernet port units; num>=1 returns the state of the corresponding port on multiple ethenet port units (SFP Fiber port is always the last counted port). On module based units, only the master module is capable of returning the external states.	yes	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	(half duplex=HD, full duplex =FD) 0=No link, 1=10MbitHD, 2=10MbitFD, 3=100MbitHD, 4=100MbitFD, 5=1000MbitFD, 7=Wlan; 8=10GbitFD
Write	void	access_right_service	not suported

CONF_ETH_LINK_TROUGHPUT

Tag code	NumDes	Message	SNMP Support
0x0a80	Message : no	Description : lower 16 bit: downlink KBPS, upper 16 bit: uplink KBPS	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	void	access_right_service	not suported

CONF_ETH_TX_PKT_BURST

Tag code	NumDes	Message	SNMP Support
0x0afe	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	eth tx pkts per burst (0:=no limitation)
Write	t_dword	access_right_service	eth tx pkts per burst (0:=no limitation)

CONF_NBR_OF_EXT_ETH_PORTS

Tag code	NumDes	Message	SNMP Support
0x0a28	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of external ethernet ports of a device
Write	void	access_right_service	not supported

CONF_NBR_OF_EXT_ETH_COPPER_PORTS

Tag code	NumDes	Message	SNMP Support
0x0a29	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of external copper ethernet ports of a device
Write	t_dword	access_right_service	returns the number of external copper ethernet ports of a device

CONF_NBR_OF_EXT_ETH_FIBER_PORTS

Tag code	NumDes	Message	SNMP Support
0x0a2a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of external fiber ethernet ports of a device
Write	t_dword	access_right_service	returns the number of external fiber ethernet ports of a device

CONF_LED_CAPS

Tag code	NumDes	Message	SNMP Support
0x0bf8	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read power/traffic LED capabilities: 0=no capabilities, 1=disengageable
Write	t_dword	access_right_service	not supported

CONF_ENABLE_TRAFFIC_LED

Tag code	NumDes	Message	SNMP Support
0x09a6	no	no	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	network activity LED (Dinion, NBC-255 camera): 0=off, 1=on
Write	t_octet	access_right_service	Enables or disables the network activity LED (Dinion, NBC-255 camera)

CONF_SND_MSS

Tag code	NumDes	Message	SNMP Support
0x0a02	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	max tcp send mss for all connections
Write	t_dword	access_right_service	set the global tcp send mss; use this to reduce the max. send segment size for all tcp connections; higher settings than default will only be used by iSCSI connections (0 means default)

CONF_EAP_IDENTITY

Tag code	NumDes	Message	SNMP Support
0x09ea	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the EAP identity
Write	p_string	access_right_service	write the EAP identity

CONF_EAP_GET_IDENTITY_LIST

Tag code	NumDes	Message	SNMP Support
0x0c4d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_minimal	not supported

Payload Structure

16	32
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	
length 2 Byte	**tag** 2 Byte
value (length - header) Bytes ...	

8 24

length

length of tagged value including length and tag field

tag

tag specifying the encoding and meaning of the value

value

value payload data (length - header bytes), can contain tagged values

Generic description

This command returns the identities used for EAP authentication. If a EAP client certificate is assigned, the command returns up to 3 identity strings. All strings are UTF-16 coded. The order of the list is the order the EAP client uses the identies. If one identity fails to authenticate the next one from the list is used.
1. The alternative subject name from the certificate (tag 0x0001)
2. The common name from the certificate subject (tag 0x0002)
3. The EAP identity from config (tag 0x0003)
Each of these entries is optional and can be missing. The complete identity list is surrounded by the tag 0x0000.
If no identity is configured or found, the tag 0x0004 is returned to indicate the empty list.

Tag Structure

16	32
length 2 Byte	**tag = 0x8000 (List entry)** 2 Byte
length 2 Byte	**tag = 0x0001 (Subject Alt Name)** 2 Byte
Alternative Subject Alt Name from certificate (length - header) Bytes ...	
length 2 Byte	**tag = 0x0002 (Common Name)** 2 Byte
Common Name from certificate (length - header) Bytes ...	
length 2 Byte	**tag = 0x0003 (Configured identity)** 2 Byte
The identity from the config (length - header) Bytes ...	
List entry may repeat	
If there are no identities the empty tag is returned. This is the only tag in the response then.	
length 2 Byte	**tag = 0x0004 (No Identity)** 2 Byte
No identity string found (nor in certificate, nor in config) 0 Bytes ...	

8 24

tag = 0x8000 (List entry)

The payload of tag list entry contains further tagged values described below.
For each identity list one list entry tag is contained in the command payload.
The highest bit (bit 15) is set to indicate that this tag will contain subtags.

tag = 0x0001 (Subject Alt Name)

Contains the Alternative Subject Name from the EAP client certificate coded as UTF-16, no zero termination.

tag = 0x0002 (Common Name)

Contains the subject's Common Name from the EAP client certificate coded as UTF-16, no zero termination.

tag = 0x0003 (Configured identity)

Contains the identity from the device config (given by the user) coded as UTF-16, no zero termination.

tag = 0x0004 (Empty list)

No identity strings were found.

CONF_EAP_ENABLE

Tag code	NumDes	Message	SNMP Support
0x09eb	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read the EAP/802.1x status (0: off/don't do EAP/802.1x; 1: on/use EAP/802.1x)
Write	t_octet	access_right_service	write the EAP/802.1x status (0: off/don't do EAP/802.1x; 1: on/use EAP/802.1x)

CONF_EAP_PASSWORD

Tag code	NumDes	Message	SNMP Support
0x09ec	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the EAP password
Write	p_string	access_right_service	write the EAP password

CONF_SWITCH_TRUNKING

Tag code	NumDes	Message	SNMP Support
0x0a6b	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	external ports bitwise members of a trunk
Write	t_octet	access_right_service	external ports bitwise members of a trunk

CONF_SWITCH_LACP_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0a6c	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	external ports bitwise enabling LACP+RSTP mode, 0: off, 1:on
Write	t_octet	access_right_service	external ports bitwise enabling LACP+RSTP mode, 0: off, 1:on

CONF_SWITCH_LACP_KEY

Tag code	NumDes	Message	SNMP Support
0x0a6d	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	LACP Key for external ports
Write	t_octet	access_right_service	LACP Key for external ports

CONF_SWITCH_RSTP_PRIO

Tag code	NumDes	Message	SNMP Support
0x0a6e	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	RSTP Priority val in range of 0 ... 61440, must be multiple of 4096
Write	t_dword	access_right_service	RSTP Priority val in range of 0 ... 61440, must be multiple of 4096

CONF_SWITCH_RSTP_HELLO_TIME

Tag code	NumDes	Message	SNMP Support
0x0a6f	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	RSTP Hello Time in range of 1...10, default 2
Write	t_octet	access_right_service	RSTP Hello Time in range of 1...10, default 2

CONF_SWITCH_RSTP_MAX_AGE

Tag code	NumDes	Message	SNMP Support
0x0a70	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	RSTP Max Age in range of 6...40, default 20
Write	t_octet	access_right_service	RSTP Max Age in range of 6...40, default 20

CONF_SWITCH_RSTP_FWD_DELAY

Tag code	NumDes	Message	SNMP Support
0x0a71	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	RSTP Forward Delay in range of 4...30, default 15
Write	t_octet	access_right_service	RSTP Forward Delay in range of 4...30, default 15

CONF_SWITCH_RSTP_VERSION

Tag code	NumDes	Message	SNMP Support
0x0a72	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	RSTP Version, 0: STP compatible, 2: RSTP
Write	t_octet	access_right_service	RSTP Version, 0: STP compatible, 2: RSTP

CONF_SWITCH_RSTP_PORT_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0a74	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	external ports bitwise enabling RSTP-only mode, 0: off, 1:on
Write	t_octet	access_right_service	external ports bitwise enabling RSTP-only mode, 0: off, 1:on

CONF_SWITCH_RSTP_PORT_EDGE

Tag code	NumDes	Message	SNMP Support
0x0a75	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	external ports bitwise determining if RSTP edge port
Write	t_octet	access_right_service	external ports bitwise determining if RSTP edge port

CONF_SWITCH_RSTP_PORT_PATHCOST

Tag code	NumDes	Message	SNMP Support
0x0a76	yes	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	RSTP pathcost of external ports in range of 1...200000000, while 0 means auto
Write	t_dword	access_right_service	RSTP pathcost of external ports in range of 1...200000000, while 0 means auto

CONF_BACKPLANE_TYPE

Tag code	NumDes	Message	SNMP Support
0x0a78	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	type of backplane 0xff: no backplane, 0xfe: old, 0: full-flavoured, 1: eco, 2: diplan
Write	%	access_right_service	%

CONF_BACKPLANE_FW_VERSION

Tag code	NumDes	Message	SNMP Support
0x0a84	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	firmware version on backplane 0: error / no (valid) backplane, else: firmware revision
Write	%	access_right_service	%

CONF_SWITCH_IGMP_IP

Tag code	NumDes	Message	SNMP Support
0x0a95	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	IP address that the VIPX1600 backplane uses as source IP in IGMP queries
Write	t_dword	access_right_service	IP address that the VIPX1600 backplane uses as source IP in IGMP queries

CONF_SWITCH_IGMP_SNOOPING_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0a9a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	state of IGMP snooping on Vitesse backplane switch
Write	t_dword	access_right_service	enable/disable IGMP snooping on Vitesse backplane switch

CONF_SWITCH_MAC_IS_SET

Tag code	NumDes	Message	SNMP Support
0x0a9d	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	determine of backplane has a valid MAC address: 0: invalid, 1: not yet known, ask later (during boot process), 2: valid
Write	%	access_right_service	%

CONF_REBOOT_SWITCH

Tag code	NumDes	Message	SNMP Support
0x0a9e	no	no	no

	Datatype	Access Level	Description
Read	%	access_right_service	%
Write	t_dword	access_right_service	reboot switch

CONF_WLAN_SSID

Tag code	NumDes	Message	SNMP Support
0x0943	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the WLAN ssid
Write	p_string	access_right_service	write the WLAN ssid

CONF_WLAN_WPA_PSK

Tag code	NumDes	Message	SNMP Support
0x0ac3	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read the WPA pre shared key
Write	p_string	access_right_service	write the WPA pre shared key

CONF_WLAN_IOCTRL

Tag code	NumDes	Message	SNMP Support
0x0c4c	no	no	no

	Datatype	Access Level	Description
Read		access_right_minimal	not supported
Write	p_octet	access_right_service	Execute ioctrl with wifi driver

CONF_PORT_FC_MODE

Tag code	NumDes	Message	SNMP Support
0x0abb	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	num=0 is the flow control mode of the external state on single ethernet port units or the internal fc mode on multiple ethernet port units; num>=1 is the fc mode of the corresponding port on multiple ethenet port units (SFP Fiber port is always the last counted port). On module based units, only the master module is capable of setting the external port fc modes.
Write	t_octet	access_right_service	num=0 is the flow control mode of the external state on single ethernet port units or the internal fc mode on multiple ethernet port units; num>=1 is the fc mode of the corresponding port on multiple ethenet port units (SFP Fiber port is always the last counted port). On module based units, only the master module is capable of setting the external port fc modes.

CONF_WLAN_SCAN

	Tag code	NumDes	Message	SNMP Support
	0x0ac6	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	Performs a WLAN network scan, see detailed description
Write		access_right_service	not supported

Payload Structure

16 32

Number of Entries 4 Bytes

Sequence of:

SSID 32 Bytes	
MAC Address 6 Bytes	
	Strength 4 Bytes
	Channel 4 Bytes
	Flags 4 Bytes

16 32

SSID

Name of Network.

MAC Address

MAC Address of AccessPoint.

Strength

Signal Strength [0;5]

Channel

Channel used by AccessPoint [0; 14]

Flags

Values:

Bit0	WEP encryption supported
Bit1	WPA encryption supported
Bit2	WPA2 encryption supported
Bit3	WPS supported

CONF_WLAN_LINK_TEST

Tag code	NumDes	Message	SNMP Support
0x0b23	no	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_service	Test WLAN setup (try ability to associate to AP with current settings and return result 0=error, 1=success)
Write		access_right_service	not supported

CONF_WLAN_OPERATING_MODE

Tag code	NumDes	Message	SNMP Support
0x0ac7	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Gets the operating mode of WLAN (0="off", 1="auto (Station only)", 2="Access point")
Write	t_octet	access_right_service	Sets the operating mode of WLAN (0="off", 1="auto (Station only)", 2="Access point")

CONF_SWITCH_POST_UPDATE_ACTION

Tag code	NumDes	Message	SNMP Support
0x0ac5	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	id that indicates which action is to be executed as the switch firmware was updated (0: none)
Write	t_octet	access_right_service	id that indicates which action is to be executed as the switch firmware was updated (0: none)

CONF_WLAN_REGION_CODE

Tag code	NumDes	Message	SNMP Support
0x0acf	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	Gets the region code for WLAN adapter (0x10="US", 0x20="CA", 0x30="EU", 0x31="SPN", 0x32="FR", 0x40="JPN", 0x41="JPN")
Write	t_word	access_right_service	Sets the region code for WLAN adapter (0x10="US", 0x20="CA", 0x30="EU", 0x31="SPN", 0x32="FR", 0x40="JPN", 0x41="JPN")

CONF_WLAN_LINK_QUALITY

Tag code	NumDes	Message	SNMP Support
0x0b1c	no	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	read WLAN link signal level
Write		access_right_service	not supported

CONF_WLAN_WPS_SETUP

Tag code	NumDes	Message	SNMP Support
0x0bc4	Message : no	Description : Reports the camera WPS PIN to be entered in the access point.	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	p_string	access_right_service	Starts a WiFi Protected Setup (WPS) session with the provided PIN (4 or 8 digits) and SSID. • PIN: The string contains the pin, a plus sign and the SSID. E.g.: "12345678+MyAccessPoint" • Push-button configuration (PBC): leave string empty • If the SSID is omitted, the device scans for a suitable access point. The status can be read via CONF_WLAN_WPS_STATUS.

CONF_WLAN_WPS_STATUS

Tag code	NumDes	Message	SNMP Support
0x0bc5	no	yes	no

	Datatype	Access Level	Description

Read	t_octet	access_right_minimal	Read the WPA configuration status. The value is a combination of the status (bit 7 to 4) and the mode (bit 3 to 0):
			• bit 7..4:
			  ○ 0 = not started
			  ○ 1 = ongoing
			  ○ 2 = successful
			  ○ 3 = rejected by the access point
			  ○ 4 = timeout
			  ○ 5 = invalid PIN
			• bit 3..0:
			  ○ 0 = none
			  ○ 1 = PBC
			  ○ 2 = PIN

Write	%	access_right_service	%

CONF_WLAN_DEFAULT_CHANNEL

Tag code	NumDes	Message	SNMP Support
0x0945	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read the WLAN channel
Write	t_octet	access_right_service	write the WLAN channel

CONF_SOCKET_KNOCKER_MODE

Tag code	NumDes	Message	SNMP Support
0x0b5c	-	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get mode of the socket knocker: 0=off, 1=on, 2=auto
Write	t_dword	access_right_service	switch socket knocker on/off: 0=off, 1=on, 2=auto

CONF_SOCKET_KNOCKER_DESTINATION

	Tag code	NumDes	Message	SNMP Support
	0x0aee	destination number (starting with 1)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get socket knocker destination and connection configuration. See detailed description for payload structure
Write	p_octet	access_right_service	specify socket knocker destination and connection configuration. See detailed description for payload structure

Payload Structure

16		32
port 2 Bytes	**ssl** 1 Byte	**nbr of sockets** 1 Byte
reserved 4 Bytes		
url ...		

port

destination port

ssl

provide ssl socket: 0=no ssl, 1=ssl

nbr of sockets

number of idle sockets to distribute. As soon as a socket is used, a new one is provided.

reserved

reserved for future use. Set to zero.

url

destination url. Zero-terminated ascii string. Currently max. 128 characters (incl. zero termination).

CONF_SOCKET_KNOCKER_DESTINATION_NAME

Tag code	NumDes	Message	SNMP Support
0x0c83	destination number (starting with 1); see CONF_SOCKET_KNOCKER_DESTINATION	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	get the 'friendly name' for a socket knocker destination url (configured by CONF_SOCKET_KNOCKER_DESTINATION). Shown to the customer to identify any connected Cloud Service. Max. 64 characters.
Write	p_string	access_right_service	set a 'friendly name' for a socket knocker destination url (configured by CONF_SOCKET_KNOCKER_DESTINATION). Shown to the customer to identify any connected Cloud Service. Max. 64 characters.

CONF_SOCKET_KNOCKER_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0b98	destination number (starting with 1)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the current status of the socket knocker. See detailed description for payload structure
Write	%	access_right_service	%

Payload Structure

16 32

state 1 Byte	**reason** 1 Byte	**reserved** 1 Byte	**reserved** 1 Byte

state

current status of socket knocker

Values:

Not running	0
Trying to connect	1
Connected (ready for cloud service)	2

reason

reason for current state

Values:

As expected	0
Unknown	1
Auto mode: DHCP off or not successful	2
Auto mode: max knocking attempts reached	3
Auto mode: max knocking time reached	4
URL could not be resolved	5
Destination not responding	6

reserved

reserved for future use.

CONF_FTP_SERVER_PORT

Tag code	NumDes	Message	SNMP Support
0x0b9f	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the Port of the camera internal FTP server for loading files to the camera; Value 0 turns server off
Write	t_word	access_right_service	set the Port of the camera internal FTP server for loading files to the camera; Value 0 turns server off

CONF_FTP_PWD

Tag code	NumDes	Message	SNMP Support
0x0b1b	SessionId (see CONF_START_CLIENT): optional parameter to identifiy a already existing ftp client session	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	gets the working directory of the configured ftp server
Write	p_string	access_right_service	not supported

CONF_ACCOUNT_LIST

Tag code	NumDes	Message	SNMP Support
0x0b1d	SessionId (see CONF_START_CLIENT): optional parameter to identifiy a already existing ftp client session	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	returns the response from the ftp/dropbox LIST command
Write	p_string	access_right_service	not supported

CONF_ACCOUNTS_CREATE_FOLDER

Tag code	NumDes	Message	SNMP Support
0x0b6c	SessionId (see CONF_START_CLIENT): optional parameter to identifiy a already existing client session	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	not supported
Write	p_octet	access_right_service	creates a new folder. Argument: folder name

CONF_ACCOUNTS_DELETE_FOLDER

Tag code	NumDes	Message	SNMP Support
0x0b6f	SessionId (see CONF_START_CLIENT): optional parameter to identifiy a already existing client session	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	not supported
Write	p_octet	access_right_service	deletes a folder. Argument: folder name

CONF_FTP_CWD

Tag code	NumDes	Message	SNMP Support
0x0b1e	SessionId: optional parameter to identifiy a already existing ftp client session	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	changes the working directory
Write	p_string	access_right_service	not supported

CONF_FTP_CDUP

Tag code	NumDes	Message	SNMP Support
0x0b1f	SessionId: optional parameter to identifiy a already existing ftp client session	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	changes to parent directory
Write	p_string	access_right_service	not supported

CONF_START_CLIENT

Tag code	NumDes	Message	SNMP Support
0x0b20	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_service	starts a ftp client session, might be needed for commands like CONF_FTP_CDUP, CONF_FTP_CWD, CONF_FTP_LIST, CONF_FTP_PWD. This session stays alive for 5 min. It can be should be shut down using CONF_FTP_STOP_CLIENT. This command returns a 2 byte sessionId that can be used to reference the session
Write	t_word	access_right_service	not supported

CONF_STOP_CLIENT

Tag code	NumDes	Message	SNMP Support
0x0b21	SessionId	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_service	not supported
Write	t_word	access_right_service	shut down the ftp session indified via the num parameter, might be needed for commands like CONF_FTP_CDUP, CONF_FTP_CWD, CONF_FTP_LIST, CONF_FTP_PWD

CONF_FTP_FILE_NAME

	Tag code	NumDes	Message	SNMP Support
	0x0b35	response: 1st OCTET inst, 2nd OCTET mode (0->rec, 1->live),	request: 1st OCTET inst, 2nd OCTET mode (0->rec, 1->live), 64 OCTETs reserved	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_service	gets the user defined ftp file name for the first recording or live instance:	
Write	p_octet	access_right_service	sets the user defined ftp file name for the first recording or live instance:	

CONF_ACCOUNT_LOGIN_TEST

	Tag code	NumDes	Message	SNMP Support
	0x0b36	yes account info (1...4)	no	no
	Datatype	**Access Level**	**Description**	
Read	t_int	access_right_user	checks if a login to the specified ftp or dropbox server is possible. In case of dropbox this command is used for authentication. FTP: return value: 0: ok , 1: Connect failed, 2: invalid user or password, 3: set directory failed, 4: general error, 5: List failed; DROPBOX: 0- Authentication session started, CONF_DROPBOX_AUTHENTICATION_STATUS message informs about the status; other value: start Authentication failed	
Write	t_int	access_right_service	not supported	

CONF_DIFF_SERV_VAL

	Tag code	NumDes	Message	SNMP Support
	0x0b27	yes 1..6 Audio/Video/Control/Alarm-Audio/Alarm-Video/Alarm-Control	no	no
	Datatype	**Access Level**	**Description**	
Read	t_octet	access_right_minimal	gets the DiffServ value for media sockets	
Write	t_octet	access_right_service	sets the DiffServ value for media sockets	

CONF_DIFF_SERV_POST_ALARM_TIME

Tag code	NumDes	Message	SNMP Support
0x0b3b	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get post alarm time (in seconds) for DiffServ alarm values
Write	t_word	access_right_service	set post alarm time (in seconds) for DiffServ alarm values. (Only if DiffServ alarm values are specified and therefore DiffServ values are changed during an alarm).

CONF_IPV4_FILTER

Tag code	NumDes	Message	SNMP Support
0x0b3c	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read the list of 2 allowed pairs of one IPv4 address and a corresponding IPv4 mask (addresses and masks in network byte order; filled completely with 0.0.0.0 entries, if disabled; in case only first entry is used, the second is filled with 0.0.0.0 / 0.0.0.0, order is address1, mask1, address2, mask2)
Write	p_octet	access_right_service	write the list of 2 allowed pairs of one IPv4 address and a corresponding IPv4 mask (addresses and masks in network byte order; filled completely with 0.0.0.0 entries, if disabled; in case only first entry is used, the second is filled with 0.0.0.0 / 0.0.0.0, order is address1, mask1, address2, mask2)

CONF_UPLINK_KBPS

Tag code	NumDes	Message	SNMP Support
0x0c03	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_service	uplink bandwidth of system in kbps
Write	t_dword	access_right_service	uplink bandwidth of system in kbps

CONF_TC_CLIENT_PARAMS

Tag code	NumDes	Message	SNMP Support
0x0c04	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	get client params: 4 bytes if client is local, 4 bytes kbps; session id is needed
Write	p_octet	access_right_user	set client params: 4 bytes if client is local, 4 bytes kbps; session id is needed

CONF_OBEY_ICMP_REDICRECTS

Tag code	NumDes	Message	SNMP Support
0x0c13	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_service	obey or ignore ICMP redirect messages, 0 = ignore, 1 = obey
Write	t_octet	access_right_service	obey or ignore ICMP redirect messages, 0 = ignore, 1 = obey

CONF_NETWORK_SERVICES

Tag code	NumDes	Message	SNMP Support
0x0c62	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns all running network services
Write	p_octet	access_right_service	returns all running network services

Payload Structure

The payload if this command is a tagged command structure. Each length field reprents the length of the field **including** the complete tag header (min length is 4 bytes).
If a tag contains subtags, this is indicated by bit 15 == 1 in tag number.

16 32

length 2 Byte	**tag 0x8000 (surrounding tag for each entry)** 2 Byte
length (always 0x0008) 2 Byte	**tag (0x0001)** 2 Byte
Numerical ID of service 4byte (32bit)	
length (text + header) 2 Byte	**tag (0x0002)** 2 Byte
Label string to display as default Num characters	
length (always 0x0008) 2 Byte	**tag (0x0003)** 2 Byte
Enable (!= 0 enabled; == 0 disabled) 4 Byte (32bit)	
Following tag is optinal. Only present if port is adjustable.	
length (always 0x0008) 2 Byte	**tag (0x0004)** 2 Byte
TCP/UDP port number 4byte (32bit)	
Next entry indicated by a 0x8000 tag. . . .	

8 24

Generic description

Reading the CONF_NETWORK_SERVICES command returns a list of TCP/UDP services which are available on the device. Each entry represents one service. The entry show if the service is enabled at the moment and what TCP/UDP port it is assigned to if a port is available. If the port cannot be adjusted the 0x0004 tag is simply missing. Client software can show this list to give the user the opportunity to check and configure all runiing network services.
Writing this command allows to configure the services returned by a preceding read. In write direction the command payload has the same structure. The client does not need to include all services returned by the read. It can only add the services to change to the write payload. The tag 0x0002 (label) must not be included in write payload. The service ID must provided and identitfies the service to change. The ID is not fixed across firmware versions and must be retrieved by a preceeding read.
Writing this command is only a request to the device to shutdown or start the services. That's why the device cannot answer with with the new state of the services. The device replies to a write operation with the actual state of the network services (the complete list). This does eventually not reflect the changes which will be done by the requests in this write operation.
ATTENTION: Most network services need a reboot of the device to be stopped or started. After writing this command the device should be rebootet!

Tag definition

tag 0x8000 (List entry)

Surrounding tag for each entry in the list. With help of the length field client software is the change to skip complete entries.

tag 0x0001 (Numerical ID)

This tag gives each service a numerical ID. The ID should be retrieved via reading this command first. You can assume, that the ID for a special service keeps constant in newer firmware versions. So this ID can be treated as a unique identifier for a specific service. When using this command in write direction to configure a service this ID must be presented to identify the service.

tag 0x0002 (Label)

A printable label for this service. If the client does not know how to translate this label it can show this label as default. The label is **not** used to identify the service and can be omitted when writing this command.

tag 0x0003 (Enable)

A value != 0 indicates that this service is enbaled and the port is open at UDP/TCP level.
A value == 0 indeicates that the service is disabled.
In write direction you can use this bit to disable a service.

tag 0x0004 (UDP/TCP port)

This tag is only present if the UDP/TCP port is adjustable for this service. Else this tag is simply missing.
For historical reasons most services set their port to 0 to show that they are disabled. If you set the 'Enable' tag to 0 at write and set the port != 0 the command will automaticly set the port to 0.
Setting 'Enabled' tag to != 0 and the port to 0 will use default port. Else the given port will be used.

CONF_CBS_COMMISSION

Tag code	NumDes	Message	SNMP Support
0x0c72	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	
Write	p_octet	access_right_minimal	Commission to CBS-Account. Action, User, Password. See detailed description for payload structure

Payload Structure

16	32
action 1 Byte	**user** ...
password ...	

action

0 = cancel commision

1 = start commission

user

user name (0-terminated, max-len 128)

password

user password (0-terminated, max-len 128)

CONF_CBS_STATUS

Tag code	NumDes	Message	SNMP Support
0x0c73	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Status of CBS Commission. see detailed Description :
Write	p_octet	access_right_minimal	update commission status, no parameters needed

Payload Structure

	16	32
cbs status 1 Byte	**cbs action status** 1 Byte	

cbs status

is a status byte for "Network->Advanced"

0 = Registered
1 = OnGoing
2 = UnRegistered
255 = Unknown

cbs action status

is a detailed status byte for "Remote Portal" dialog

0 = Ok
1 = OnGoing
2 = UnRegistered
3 = Canceled
0xe0 = ReconfigurationForbidden
0xe1 = ReconfigurationConflict
0xe2 = crsaReconfigurationUnprocessibleEntity
0xe3 = crsaReconfigurationLocked
0xe4 = crsaReconfigurationServiceUnavalibale
0xe5 = crsaReconfigurationAuthError
0xf9 = crsaDeviceSecretError
0xfa = crsaReceiveError
0xfb = crsaMessageError
0xfc = crsaInvalidMacAddressFormat
0xfd = crsaFriendlyNameEmpty
0xfe = crsaCloudUnreachable
0xff = crsaUnexpectedError

CONF_CBS_DESTINATION

	Tag code	NumDes	Message	SNMP Support
	0x0c75	destination number (starting with 1)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get CBS destination and connection configuration. See detailed description for payload structure
Write	p_octet	access_right_service	set CBS destination and connection configuration. See detailed description for payload structure

Payload Structure

16		32
port 2 Bytes	**ssl** 1 Byte	**reserved1** 1 Byte
reserved2 4 Bytes		
url ...		

port

destination port

ssl

provide ssl socket: 0=no ssl, 1=ssl

reserved1

reserved for future use. Set to zero.

reserved2

reserved for future use. Set to zero.

url

destination url. Zero-terminated ascii string. Currently max. 128 characters (incl. zero termination).

example payload for port 443, ssl 1, url "api.remote.boschsecurity.com"

0x01bb010000000000006170692e72656d6f74652e626f736368736563757269747 92e636f6d00

example payload for port 443, ssl 1, url "api.remotest.cbs.boschsecurity.com"

0x01bb010000000000006170692e72656d6f746573742e6362732e626f736368736563757269747 92e636f6d00

CONF_MULTICAST_VIDEO_GROUP_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x01b4	abs video coder instance	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	read the multicast video group address	
Write	p_string	access_right_service	set the multicast video group address (range: 224.0.0.10 .. 239.255.255.255)	

CONF_MULTICAST_VIDEO_GROUP_IP

	Tag code	NumDes	Message	SNMP Support
	0x01b1	abs video coder instance	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	read the multicast video group address	
Write	t_dword	access_right_service	set the multicast video group address (range: 224.1.0.0 .. 239.255.255.255)	

CONF_MULTICAST_AUDIO_GROUP_IP_STR

	Tag code	NumDes	Message	SNMP Support
	0x0cd2	abs audio coder instance	no	no
	Datatype	**Access Level**	**Description**	
Read	p_string	access_right_minimal	read the multicast audio group address	
Write	p_string	access_right_service	set the multicast audio group address (range: 224.0.0.10 .. 239.255.255.255)	

CONF_MULTICAST_AUDIO_GROUP_IP

Tag code	NumDes	Message	SNMP Support
0x0cd1	abs audio coder instance	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the multicast audio group address
Write	t_dword	access_right_service	set the multicast audio group address (range: 224.1.0.0 .. 239.255.255.255)

CONF_MULTICAST_VIDEO_PORT_STR

Tag code	NumDes	Message	SNMP Support
0x0288	coder instance	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the multicast video UDP port number
Write	p_string	access_right_service	set the multicast video UDP port (even port numbers only)

CONF_MULTICAST_VIDEO_PORT

Tag code	NumDes	Message	SNMP Support
0x0286	coder instance	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the multicast video UDP port number
Write	t_word	access_right_service	set the multicast video UDP port (even port numbers only)

CONF_MULTICAST_AUDIO_PORT_STR

Tag code	NumDes	Message	SNMP Support
0x01b5	coder instance	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the g711 multicast UDP port number
Write	p_string	access_right_service	set the multicast audio G711 UDP port (even port numbers only)

CONF_MULTICAST_AUDIO_PORT

Tag code	NumDes	Message	SNMP Support
0x01b2	coder instance	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	read the g711 multicast UDP port number
Write	t_word	access_right_service	set the multicast audio G711 UDP port (even port numbers only)

CONF_MULTICAST_TTL

Tag code	NumDes	Message	SNMP Support
0x0267	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get the TTL (time to live) in IP-Header for multicast packets
Write	t_octet	access_right_service	set the TTL (time to live) in IP-Header for multicast packets

CONF_IGMP_VERSION

Tag code	NumDes	Message	SNMP Support
0x09e5	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read the igmp(internet group management protocol) version (0: Automatic from network; 1: Version1; 2: Version2; 3: Version3)
Write	t_octet	access_right_service	Set the igmp version (0: Automatic from network; 1: Version1; 2: Version2; 3: Version3).

CONF_STREAMING_VAL

Tag code	NumDes	Message	SNMP Support
0x01b9	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	=0: set multicast streaming mode to off; =1: set multicast streaming mode mpeg4 to on; =2: set multicast streaming mode mpeg2 to on; =3: set multicast streaming mode mpeg4+mpeg2 to on
Write	t_octet	access_right_service	=0: set multicast streaming mode to off; =1: set multicast streaming mode mpeg4 to on; =2: set multicast streaming mode mpeg2 to on; =3: set multicast streaming mode mpeg4+mpeg2 to on

CONF_VIDEO_ENC_STREAMING

Tag code	NumDes	Message	SNMP Support
0x099a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	video streaming on coder (video encoder n: coderbits=1<<(n-1))
Write	t_dword	access_right_service	enable video streaming on coder. (video encoder n: coderbits=1<<(n-1))

CONF_AUDIO_ENC_STREAMING

Tag code	NumDes	Message	SNMP Support
0x099b	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	audio streaming on coder (audio encoder n: coderbits=1<<(n-1))
Write	t_dword	access_right_service	enable audio streaming on coder (audio encoder n: coderbits=1<<(n-1))

CONF_START_MULTICAST_STREAMING

Tag code	NumDes	Message	SNMP Support
0x0b8b	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	video streaming on coder (1st coder-> 1st octet, 2nd coder-> 2nd octet....)
Write	p_octet	access_right_service	enable video streaming on coder. 1st Coder->1 octet, 2nd Coder ->2nd octet....

CONF_AUDIO_STREAMING_ENCODING

Tag code	NumDes	Message	SNMP Support
0x0b8d	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	defines the audio encoding that is used for streaming (1: g711, 2: aac)
Write	t_dword	access_right_service	defines the audio encoding that is used for streaming (1: g711, 2: aac)

CONF_JOIN_STREAM_NOW

Tag code	NumDes	Message	SNMP Support
0x01d4	no	no	no

	Datatype	Access Level	Description
Read	void	access_right_minimal	not supported
Write	flag	access_right_live	start joining a multicast stream from the alarm IP

CONF_DYNDNS_SERVER

Tag code	NumDes	Message	SNMP Support
0x030f	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete - IP address of VCS dynamic DNS server
Write	t_dword	access_right_service	obsolete - IP address of VCS dynamic DNS server

CONF_DYNDNS_TIMEOUT

Tag code	NumDes	Message	SNMP Support
0x0310	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete - time in sec when the VJ should retrigger its VCS dynamic server registration (30 .. 86400)
Write	t_dword	access_right_service	obsolete - time in sec when the VJ should retrigger its VCS dynamic server registration (30 .. 86400)

CONF_DYNDNS_SERVER_REPLY

Tag code	NumDes	Message	SNMP Support
0x0311	no	yes	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	not supported; will generate message when server answers
Write	p_string	access_right_service	not supported; will generate message when server answers

CONF_DYNDNS_HOST_NAME

Tag code	NumDes	Message	SNMP Support
0x0a56	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the dyndns host name that is to be registered at dyndns.com
Write	p_string	access_right_service	write the dyndns host name that is to be registered at dyndns.com

CONF_DYNDNS_USER_NAME

Tag code	NumDes	Message	SNMP Support
0x0a57	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the user name of account at dyndns.com
Write	p_string	access_right_service	write the user name of account at dyndns.com

CONF_DYNDNS_PASSWORD

Tag code	NumDes	Message	SNMP Support
0x0a58	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read the password of account at dyndns.com
Write	p_string	access_right_service	write the password of account at dyndns.com

CONF_DYNDNS_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0a59	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	1: enable, 0: disable registering at dyndns.com
Write	t_octet	access_right_service	enable/disable registering at dyndns.com

CONF_DYNDNS_STATE

Tag code	NumDes	Message	SNMP Support
0x0a5a	no	yes	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	state of registering at dyndns.com, see detailed description
Write	%	access_right_service	%

Payload Structure

State
1 Byte

8

State

the current state of the process of registering the device at dyndns.com

Values:

successfully updated	0
updated with unchanged cfg	1
dyndns username or password wrong	2
not fully qualified domain name (e.g. not in form host.dyndns.org)	3
wrong host name for this dyndns user account	4
too many hosts in last update	5
host name is blocked by dyndns for update abuse	6
no user agent submitted or http method not permitted	7
dyndns server error dns related	8
dyndns server error maintenance related	9
update not done due to misconfiguration	252
fatal error during update process	253
update in process	254
switched off	255

CONF_DYNDNS_LAST_REGISTER

Tag code	NumDes	Message	SNMP Support
0x0a5b	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	data of last successful registration at dyndns.com, 8 bytes, first DWORD: register time in secs since 2000, second DWORD: last registered IP
Write	p_octet	access_right_service	data of last successful registration at dyndns.com, 8 bytes, first DWORD: register time in secs since 2000, second DWORD: last registered IP

CONF_DYNDNS_FORCE_REGISTER_NOW

Tag code	NumDes	Message	SNMP Support
0x0a5c	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	writing forces registering at dyndns.com / reading checks if forcing is allowed (has not been done since bootup)
Write	t_octet	access_right_service	writing forces registering at dyndns.com / reading checks if forcing is allowed (has not been done since bootup)

CONF_DYNDNS_PROVIDER

Tag code	NumDes	Message	SNMP Support
0x0b67	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read selected dyndns provider (0: dyndns.org, 1: no-ip.com, 2: selfhost.de)
Write	t_octet	access_right_service	write selected dyndns provider (0: dyndns.org, 1: no-ip.com, 2: selfhost.de)

CONF_DYNDNS_STATUS_MAIL_ONOFF

Tag code	NumDes	Message	SNMP Support
0x0b71	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read enable or disable state for sending status update emails on Dyndns status change (0: off (default), 1: on)
Write	t_octet	access_right_service	enable or disable sending status update emails on Dyndns status change (0: off (default), 1: on)

CONF_DYNDNS_MAIL_SMTP_SRV_IP_STR

Tag code	NumDes	Message	SNMP Support
0x0b72	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read address of SMTP server to be used for for Dyndns status mail transmission
Write	p_string	access_right_service	set address of SMTP server to be used for for Dyndns status mail transmission

CONF_DYNDNS_MAIL_SMTP_LOGIN

Tag code	NumDes	Message	SNMP Support
0x0b73	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read login for SMTP server to be used for for Dyndns status mail transmission
Write	p_string	access_right_service	set login for SMTP server to be used for for Dyndns status mail transmission

CONF_DYNDNS_MAIL_SMTP_PASS

Tag code	NumDes	Message	SNMP Support
0x0b74	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read password for SMTP server to be used for for Dyndns status mail transmission
Write	p_string	access_right_service	set password for SMTP server to be used for for Dyndns status mail transmission

CONF_DYNDNS_MAIL_DEST_EMAIL_ADDR

Tag code	NumDes	Message	SNMP Support
0x0b75	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read destination email address to be used for for Dyndns status mail transmission
Write	p_string	access_right_service	set destination email address to be used for for Dyndns status mail transmission

CONF_DYNDNS_MAIL_SENDER_NAME

Tag code	NumDes	Message	SNMP Support
0x0b76	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read senders name to be used for for Dyndns status mail transmission
Write	p_string	access_right_service	set senders name to be used for for Dyndns status mail transmission

CONF_DYNDNS_MAIL_TEST_SEND

Tag code	NumDes	Message	SNMP Support
0x0b77	no	no	no

	Datatype	Access Level	Description
Read	%	access_right_minimal	%
Write	t_octet	access_right_service	send test mail with current Dyndns status to configured destination

CONF_AUTODETECT_REPLY_GROUP

Tag code	NumDes	Message	SNMP Support
0x0956	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the multicast group to which (when set) the VJ will listen for multicast autodetect requests
Write	t_dword	access_right_service	set the multicast group to which (when set) the VJ will listen for multicast autodetect requests on port 1800; when set to an invalid multicast address, the default address of 225.86.67.83 is used

CONF_UNSOLICITED_AUTODETECT_REPLY_TIME

Tag code	NumDes	Message	SNMP Support
0x0957	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the time in seconds, when the VJ shall send out unsolicited autodetect reply to the RCP port(off when set to 0)
Write	t_dword	access_right_service	set the time in seconds, when the VJ shall send out unsolicited autodetect reply to port 1800 (off when set to 0); uses broadcast, and multicast if AUTODETECT_REPLY_GROUP is set

CONF_DISCOVER_PORT

Tag code	NumDes	Message	SNMP Support
0x0976	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the discover port
Write	t_word	access_right_service	set the discover port

CONF_LOCAL_HTTP_PORT

Tag code	NumDes	Message	SNMP Support
0x0954	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the local HTTP port for browser access
Write	t_word	access_right_service	set the local HTTP port for browser access (NOTE: it is not allowed to turn off both, HTTP and HTTPS, at the same time)

CONF_LOCAL_HTTPS_PORT

Tag code	NumDes	Message	SNMP Support
0x0a0e	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the local HTTPS port for browser access
Write	t_word	access_right_service	set the local HTTPS port for browser access (NOTE: it is not allowed to turn off both, HTTP and HTTPS, at the same time)

CONF_RCP_SERVER_PORT

Tag code	NumDes	Message	SNMP Support
0x0a17	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the local RCP server TCP port number
Write	t_word	access_right_service	set the local RCP server TCP port number, allowed: 0 or 1756 (Reboot necessary)

CONF_TELNET_PORT

Tag code	NumDes	Message	SNMP Support
0x0a18	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	get the local Telnet TCP port number
Write	t_word	access_right_service	set the local Telnet TCP port number

CONF_RTSP_PORT

Tag code	NumDes	Message	SNMP Support
0x0a63	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	reads the LocalRtspPort
Write	t_dword	access_right_service	sets the local Rtsp port (0: Rtsp off)

CONF_ENABLE_UPNP

Tag code	NumDes	Message	SNMP Support
0x0ade	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	0: Upnp disabled; 1: Upnp enabled
Write	f_flag	access_right_service	0: Upnp disabled; 1: Upnp enabled

CONF_ENABLE_ONVIF

Tag code	NumDes	Message	SNMP Support
0x0bbd	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	0: Onvif disabled; 1: Onvif enabled (default value)
Write	f_flag	access_right_service	0: Onvif disabled; 1: Onvif enabled (default value)

CONF_GET_RTSP_SESSION_ID

Tag code	NumDes	Message	SNMP Support
0x0ae8	random value from (Rtsp Session setup)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_live	gets the rcp session id of the rtsp session (identified by the random value)
Write	t_dword	access_right_service	not supported

CONF_HSTS_ENABLED

Tag code	NumDes	Message	SNMP Support
0x0c07	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the HSTS state (0=OFF, 1=ON 2=on + Http port redirect to HTTPS when HTTP port is also set to 0
Write	t_dword	access_right_service	set HSTS state (0=OFF, 1=ON 2=on + Http port redirect to HTTPS when HTTP port is also set to 0

CONF_ALLOW_BASIC_HTTP_AUTH_ON_NON_SSL_SOCK

Tag code	NumDes	Message	SNMP Support
0x0cd8	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	allow basic http authentication on non SSL sockets, 0: not allowed (default), 1: allowed
Write	f_flag	access_right_service	allow basic http authentication on non SSL sockets, 0: not allowed (default), 1: allowed

CONF_PROTECT_HTTP_COOKIE

Tag code	NumDes	Message	SNMP Support
0x0cdc	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	0 = Don't protect default session cookie with HttpOnly (legacy behaviour), 1 = Mark default session cookie with HttpOnly flag
Write	f_flag	access_right_service	0 = Don't protect default session cookie with HttpOnly (legacy behaviour), 1 = Mark default session cookie with HttpOnly flag

CONF_TCP_FWD

Tag code	NumDes	Message	SNMP Support
0x0b2f	yes (Forwarder index 1...max, max=4: devices, max=32: generic)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	reads the Settings of the tcp forwarder per device (WORD http listener port, WORD http forwarder port, WORD https listener port, WORD https forwarder port, string: ip)
Write	p_octet	access_right_service	writes the Settings of the tcp forwarder per device (WORD http listener port, WORD http forwarder port, WORD https listener port, WORD https forwarder port, string: ip)

CONF_PREPARE_FOR_RECORDING

Tag code	NumDes	Message	SNMP Support
0x0b51	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported, see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

rmt dev idx 16 Bits		rec idx 16 Bits	
flags 8 Bits	**storage/lun cnt** 8 Bits	**strg/lun idx 1** 8 Bits	N x 8 Bits

strg/lun idx N 8 Bits

rmt dev idx

index of a remote device entry configured by CONF_ADD_REMOTE_DEVICE from 1 to n. Value 0 means the local device.

rec idx

recording index

Values:

primary recording	1
secondary recording	2

flags

command specific flags

Values:

remote managed storage	0x01

storage/lun cnt

number of following index enties for storages/luns (max. 8)

strg/lun idx 1 - N

this is the index of an as managed configured storage or the lun index of local storage of the remote device. (1 - n, n = storage/lun cnt)

CONF_PREPARE_FOR_RECORDING

This command is used to prepare a remote device for recording via a another device (e.g remote recording on a transcoder). Normally the remote device will be configured to record on one or more (upto 8) local managed storage, which is already configured by the command CONF_STORAGE_LIST. The Storage can be a local storage on the remote device or any storage on the local device. This is choosen by the flag "remote managed storage". If this flag is set, the storage/lun idx refers the lun index of a local storage on the remote device. Otherwise it is the index from the storage list of the local device. The preparation will be the configuration of the micro vrm on the recording device and in case of remote recording constellation, it will allow the remote recording on the storage managing device.

specific errors

0x01	no local storage configured
0x02	remote dev config error
0x03	local dev config error
0x04	remote dev offline
0x05	common errors
0x06	remote dev vrm managed
0x07	iscsi auth config error

CONF_ADD_REMOTE_DEVICE

	Tag code	NumDes	Message	SNMP Support
	0x0b52	entry index(1...4)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_live	read entry of remote devices, see detailed description	
Write	p_octet	access_right_service	add entry of remote devices, see detailed description	

Payload Structure

mac ...	
mac 6 bytes	**reserved** 2 bytes
line cnt 16 Bits	**max coder per line** 16 Bits
nbr of port entries 32 Bits	
forwarder entry 1 4 Bytes	
. . .	
forwarder entry N 4 Bytes	
url len 16 Bits	**url** ...

url (url len) bytes ...

pwd len 16 Bits	**pwd** ...

pwd (pwd len) bytes ...	
device name len 16 Bits	**device name** ...

device name (device name len) bytes ...

mac

mac of the device (will be stored only as information in config)

line cnt

max lines of the device (will be stored only as information in config)

max coder per line

max number of coder instances per line (will be stored only as information in config)

nbr of port entries

number of forward port entries following this field (actual limited to 2)

forwarder entry 1 - N

forwarder port entry (see payload description)

url len

length of the following url (actual limit is 64)

url

url for the connection to the remote device, ascii string including zero termination

pwd len

length of the following password (actual limit is 64)

pwd

password for the connection, ascii string including zero termination

device name len

length of the following device name (actual limit is 64)

device name

device name of the remote device, ascii string including zero termination

forwarder entry

16	32
remote port 16 Bits	**forwarder port** 16 Bits
8	24

remote port

port of the remote device

forwarder port

local port which forwards all data to the remote port

ADD_REMOTE_DEVICE

Adds a device with forwarder ports, all data to/from the forwarder ports will be forwarded to/from the configured device port. It is allowed to store complete zeroed forwarder/remote port entries. For entries which are referenced by CONF_RCP_CONNECT_SALVO command num param, it it nessessary, that the first forward port entry contains the http port and the second one the https port. If one of these ports shall not be used, at least a zeroed entry (remote/forwarder port to zero) has to be placed. The max lines and coder per lines are just information, which will be stored together in the config. To delete an entry, send an empty or zeroed payload.

CONF_UPNP_SEARCH_IP_CONN_SERVICE

Tag code	NumDes	Message	SNMP Support
0x0b59	if set to 0xFFFF a extended payload is returned)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	searches for devices that support the upnp WANIPConnection service. Returns a list of devices entries (64 bytes addr, 64 bytes name, opt. 64 byte unique id (if extended payload is requested))
Write	p_octet	access_right_service	not supported

CONF_UPNP_TCP_FWD

Tag code	NumDes	Message	SNMP Support
0x0b5a	index of the remote device (1...4)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_service	configures port forwarding in a upnp device that support the WANIPConnection:1 service. Input 64 bytes: address of the WANIPConnection:1 service in the router (returned by CONF_UPNP_SEARCH_IP_CONN_SERVICE).

CONF_DHCP_VAL

Tag code	NumDes	Message	SNMP Support
0x00af	no	no	yes

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read the DHCP state (0=OFF, 1=ON, 3=ON+APIPA-Fallback);
Write	t_octet	access_right_service	set DHCP ip configuration value (0=OFF, 1=ON, 2=ON, but do not kick mechanism now, 3=ON+APIPA-Fallback)

CONF_DHCP_ON

Tag code	NumDes	Message	SNMP Support
0x00ad	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	read the DHCP ON state
Write	f_flag	access_right_service	enable DHCP ip configuration

CONF_DHCP_OFF

Tag code	NumDes	Message	SNMP Support
0x00ae	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_minimal	read the DHCP OFF state
Write	f_flag	access_right_service	disable DHCP ip configuration

CONF_DHCP_STABLE

Tag code	NumDes	Message	SNMP Support
0x0ac8	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	read whether DHCP machine is in possession of an IP
Write	%	access_right_service	%

CONF_DHCP_COMPLIANCY

Tag code	NumDes	Message	SNMP Support
0x0ada	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	mode of dhcp operation, 0: use of last-assigned ip during non-available dhcp server allowed; 1: no fallback ips used, more compliant to rfc 2131
Write	t_octet	access_right_service	mode of dhcp operation, 0: use of last-assigned ip during non-available dhcp server allowed; 1: no fallback ips used, more compliant to rfc 2131

CONF_SNMP_SRV_PORT

Tag code	NumDes	Message	SNMP Support
0x0a25	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	Read the snmp server port
Write	t_word	access_right_service	Set the snmp server port (reset nescessary)

CONF_SNMP_TRAPS_HOST

Tag code	NumDes	Message	SNMP Support
0x00b6	yes	no	yes

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	not documented

CONF_SNMP_TRAPS_HOST_STR

Tag code	NumDes	Message	SNMP Support
0x00b7	yes	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	not documented
Write	p_string	access_right_service	not documented

CONF_NBR_OF_TRAPS_HOSTS

Tag code	NumDes	Message	SNMP Support
0x029d	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Number of hosts snmp traps can be sent to
Write	t_dword	access_right_service	read only

CONF_SNMP_TRAP_LIST

Tag code	NumDes	Message	SNMP Support
0x0a11	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	List of commands to be posted as snmp trap. See detailed description.
Write	p_octet	access_right_service	List of commands to be posted as snmp trap. see detailed description

Payload Structure

Trap Descriptor 1 4 Bytes	
N x 4 Bytes	**Trap Descriptor N** 4 Bytes ...

Trap Descriptor 1 - N

Sequence of:

16	32
RCP Message Code 2 Bytes	**Flags** 2 Bytes
8	24

RCP Msg Code

The code of the RCP message that is to be forwarded as SNMP trap.

Bitmask of Flags

Values:

SNMP_TRAP_ENABLED	1

CONF_PMPP_PORT

Tag code	NumDes	Message	SNMP Support
0x0aba	no	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	gets the Pmpp Port of the Pmpp server(0: off)
Write	t_word	access_right_service	sets the Pmpp Port of the pmpp server(0: off)

CONF_PMPP_ADDRESS

Tag code	NumDes	Message	SNMP Support
0x0ab9	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	gets the Pmpp Address (allowed values: 1-63;)
Write	t_octet	access_right_service	sets the Pmpp Address (allowed values: 1-63;)

CONF_SYSCONTACT

Tag code	NumDes	Message	SNMP Support
0x00ba	no	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the system contact required in SNMP
Write	p_string	access_right_service	write the system contact required in SNMP

CONF_SYSLOCATION

Tag code	NumDes	Message	SNMP Support
0x00bb	no	no	yes

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	read the system location required in SNMP
Write	p_string	access_right_service	write the system location required in SNMP

CONF_SNMP_READ_COMMUNITY

Tag code	NumDes	Message	SNMP Support
0x0b16	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read the community string for snmp read access
Write	p_string	access_right_service	write the community string for snmp read access

CONF_SNMP_WRITE_COMMUNITY

Tag code	NumDes	Message	SNMP Support
0x0b17	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read the community string for snmp write access
Write	p_string	access_right_service	write the community string for snmp write access

CONF_SNMP_TRAP_COMMUNITY

Tag code	NumDes	Message	SNMP Support
0x0b18	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_service	read the community string used in snmp traps
Write	p_string	access_right_service	write the community string used in snmp traps

CONF_SNMP_SERVER_MODE

Tag code	NumDes	Message	SNMP Support
0x0c4e	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_service	Read the snmp server mode (0=v1 legacy mode; 1=v1; 8=v3; 9=v1+v3)
Write	t_octet	access_right_service	Write the snmp server mode (0=v1 legacy mode; 1=v1; 8=v3; 9=v1+v3); Support for options 1=v1 and 9=v1+v3 removed in V6.50 and later

CONF_SNMP_USER_PROFILE

Tag code	NumDes	Message	SNMP Support
0x0c4f	yes 1=(static bound to local service user profile and its auth-password); 3=(profile for a remote account sending SNMP traps to)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	Read a snmp (non-legacy) mode user profile
Write	p_octet	access_right_service	Write a snmp (non-legacy) mode user profile

Payload Structure

16 32

UserName 32 Bytes ...			
AuthPass 32 Bytes ...			
PrivPass 32 Bytes ...			
AuthMode 1 Byte	**PrivMode** 1 Byte	**SnmpVersion** 1 Byte	**IsTrapUser** 1 Byte
Profile 1 Byte	**Reserved11** 1 Byte	**Reserved12** 1 Byte	**Reserved13** 1 Byte
Reserved2 4 Bytes			
Reserved3 4 Bytes			
Reserved4 4 Bytes			

8 24

UserName

Zero terminated string with user name. Or zero to delete this user entry.

Values:

AuthPass

Zero terminated string with authentication password.

Values:

PrivPass

Zero terminated string with privacy password.

Values:

AuthMode

Authentication mode

Values:

No authentication	1
MD5 authentication	2
SHA1 authentication	3

PrivMode

Privacy mode

Values:

No privacy	1
DES privacy	2
AES privacy	4

SnmpVersion

SNMP version for this user

Values:

SNMP Version 3	3

IsTrapUser

Define if this account is a local account or a account to sent traps to

Values:

Local account	0
Remote trap account	1

Profile

Select one of the access rights profiles

Values:

Read only profile	0
Read write profile	1

Reserved11

Should be zero

Values:

Reserved12

Should be zero

Values:

Reserved13

Should be zero

Values:

Reserved2

Should be zero

Values:

Reserved3

Should be zero

Values:

Reserved4

Should be zero

Values:

CONF_STREAMING_GATEWAY_CONFIG

Tag code	NumDes	Message	SNMP Support
0x0b24	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read streaming gateway configuration, see detailed description
Write	p_octet	access_right_service	write streaming gateway configuration, see detailed description

Payload Structure

gateway line 16 Bits		**gateway stream** 8 Bits	**action** 8 Bits
protocol 8 Bits	**version** 8 Bits	**connection** 8 Bits	**reserved** 8 Bits
manufacturer id 32 Bits			
device type 32 Bytes ...			
...			
optional parameters 1 M Bytes			

N x M Bytes	**optional parameters N** M Bytes ...

gateway line

local line of streaming gateway id

gateway stream

local stream of streaming gateway starting with 0 for the first stream

action

action for write direction, choose between add camera and remove camera, in case of remove camera, except this field, "gateway line", "gateway stream" and "version" all other fields in the payload are irrelevant

Values:

add camera	0
remove camera	1

protocol

choose protocol type

Values:

none	0
onvif	1
jpeg	2
bosch	3
rtsp	4
rtsp h263	5
rtsp h264	6
rtsp jpeg	7

version

version of the command, current version is 1

connection

connection type

Values:

udp unicast	0
udp multicast	1
tcp	2

manufacturer id

distinguish between bosch cameras and foreign cameras

Values:

unknown	0

bosch	1

device type

max 32 ascii characters including zero termination

optional parameters 1 - N

optional additional parameters, if nessessary depends on the camera

STREAMING_GATEWAY_CONFIG

This command is used to read and write the configuration of the streaming gateway. For the read direction only "gateway line" and "gateway stream" parameter are relevant for the send diretion. The reply payload will contain the full parameter set. For write direction use the full parameter set to configure a streaming gate camera, if the action is "add camera". Depending on the camera type and connection type, additional optional parameters are nessessary. E.g. the device simulator is able to connect to another bosch camera or device specified by the url ([ip]:[port]/[line (1,2,...)]/[coder idx (1,2,...)]). E.g. with url "10.1.10.20:80/1/1" it connects to the device on ip 10.1.10.20 and port 80 on the first line and the first video encoder.

Payload Structure for optional parameters

16 32

type 16 Bits	**size** 16 Bits
payload N Bytes ...	
...	

8 24

optional parameters

the optional parameters contain additional parameters. Depending on the camera type and connection, these parameters may be nessessary (e.g. url for connecting a remote device).

type

type of the optional parameters

Values:

none	0
url	1
stream	2
camera token	3
profile token	4

size

size in bytes of the following payload region excluding the "type" and "size" field. Has to be 4 byte aligned.

payload

payload for the additional parameter, structure depends on the "type"

url

zero terminated ascii string containing the url of the remote camera. Length of the string including zero termination is limited by the "length" of the optional parameter

stream

remote camera line	16 bit network order
remote camera stream	8 bit
reserved	8 bit

camera token

see detailed description for camera token

profile token

see detailed description for profile token

CONF_STREAMING_GATEWAY_ACTIVE_LINES

Tag code	NumDes	Message	SNMP Support
0x0b25	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the number of configured streaming gateway lines
Write	t_dword	access_right_service	not supported

CONF_STREAMING_GATEWAY_MAX_LINES

Tag code	NumDes	Message	SNMP Support
0x0b26	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the max number of configurable streaming gateway lines
Write	t_dword	access_right_service	not supported

CONF_ISCSI_IP

Tag code	NumDes	Message	SNMP Support
0x09aa	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	deprecated in fw > 4.00
Write	t_dword	access_right_service	deprecated in fw > 4.00

CONF_ISCSI_PORT

Tag code	NumDes	Message	SNMP Support
0x09ab	yes (since fw > 4.00)	no	no

	Datatype	Access Level	Description
Read	t_word	access_right_minimal	numdes 0: connect port, 1: tunnel port, 2: server port
Write	t_word	access_right_service	numdes 0: connect port, 1: tunnel port, 2: server port

CONF_ISCSI_LUN

Tag code	NumDes	Message	SNMP Support
0x09ac	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	deprecated in fw > 4.00
Write	t_dword	access_right_service	deprecated in fw > 4.00

CONF_ISCSI_TARGET_IDX

Tag code	NumDes	Message	SNMP Support
0x09f9	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	deprecated in fw > 4.00
Write	t_dword	access_right_service	deprecated in fw > 4.00

CONF_ISCSI_TARGET

Tag code	NumDes	Message	SNMP Support
0x09ad	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	gets the iscsi target name string
Write	p_string	access_right_service	deprecated in fw > 4.00

CONF_ISCSI_TCP_CONNECTIONS

Tag code	NumDes	Message	SNMP Support
0x09ae	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the number of concurrent tcp connections to iscsi target
Write	t_dword	access_right_service	set the number of concurrent tcp connections to iscsi target

CONF_ISCSI_DISCOVERY

Tag code	NumDes	Message	SNMP Support
0x09cc	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	performs a discovery and returns the result in an XML-like string; parameter ip (DWORD) and pwd string (64 char) structure in p_octet
Write	void	access_right_service	not supported

CONF_ISCSI_TARGET_PWD

Tag code	NumDes	Message	SNMP Support
0x09ce	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_user	set the password to authenticate at the iSCSI server. FW version >= 4.00: This command sets the password of the first entry of the CONF_ISCSI_AUTH list. If all characters of the string are the '*', the old stored value is not replaced. deprecated! use CONF_ISCSI_AUTH
Write	p_string	access_right_service	get the password to authenticate at the iSCSI server. FW version >= 4.00: This command returns the password of the first entry of the CONF_ISCSI_AUTH list. deprecated! use CONF_ISCSI_AUTH

CONF_ISCSI_LOCK_OVERRIDE

Tag code	NumDes	Message	SNMP Support
0x09d2	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	see detailed description
Write	p_string	access_right_service	see detailed description

Payload

A String with the following structure - target_id:target_idx:lun

This string identifies the iscsi lun on which the lock is to be overwritten. This option clears its value after a successful override.

Note

The datatype of this command was FLAG and changed to STRING with fw 2.50. The target_id needs o be resolved by rules configured by CONF_TARGET_ID_RESOLVE_RULES. In older version the target id was the ipv4 address, which is now the default rule for target id resolving. The string format of the target id is the same as of the ipv4.

CONF_ISCSI_LOCK_RELEASE_ON_LEAVE

Tag code	NumDes	Message	SNMP Support
0x09e4	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	returns wheather the next iSCSI logout should use the release action
Write	%	access_right_service	%

CONF_ISCSI_INITIATOR_NAME

Tag code	NumDes	Message	SNMP Support
0x09d8	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	returns the used initiator name; only applicable when iSCSI is connected
Write	void	access_right_service	not supported

CONF_ISCSI_INITIATOR_NAME_EXTENTION

Tag code	NumDes	Message	SNMP Support
0x09d9	no	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	returns the used initiator name extention; used for identification only
Write	p_string	access_right_service	set the used initiator name extention; used for identification only

CONF_ISCSI_SERVER_STATE

Tag code	NumDes	Message	SNMP Support
0x0a2b	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the state of the iscsi server (0: server down, 1: server running)
Write	t_dword	access_right_service	set the iscsi server state (0: shutdown server, 1: start server)

NOTE

The iscsi server is also started or stopped when you write the command CONF_STORAGE_LIST. If the storage list is empty or only contains entries with the 'iSCSI export' field set to 'For local use only' (0x00), the server is stopped if it is running cause no targets and luns would be provided. If the list contains at least one entry with the 'iSCSI export' field set to 'Make storage available through iSCSI' (0x01), the server is started, if not already running, to provide the iscsi service.

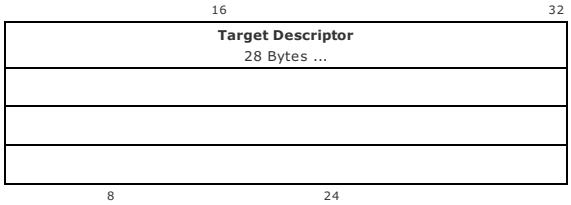
CONF_ISCSI_MNI

Tag code	NumDes	Message	SNMP Support
0x0aa0	no	yes	no

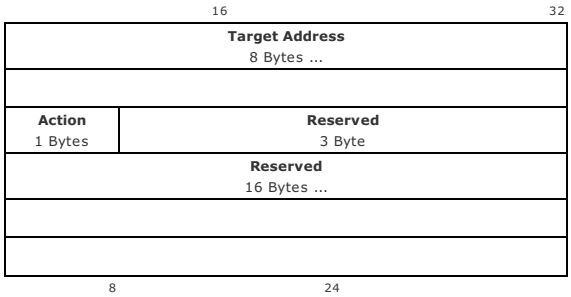
	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	monitor iscsi targets.
Write	p_octet	access_right_service	monitor iscsi targets.

Request Payload Structure

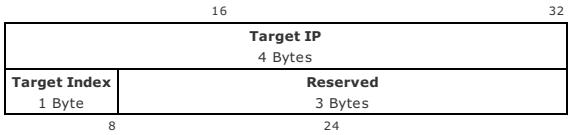
Sequence of (max 64):



Target Descriptor



Target Address



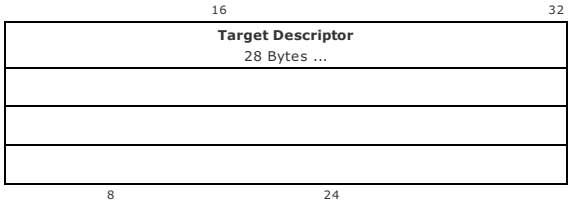
Action

Values:

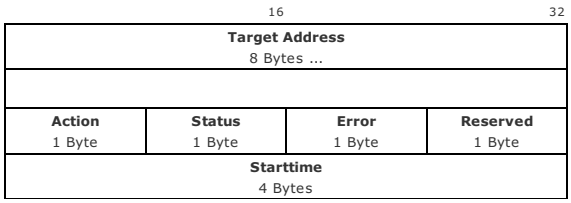
DELETE	0x00
ADD	0x01

Response Payload Structure

Sequence of (max 64):



Target Descriptor



Session Status 1 Byte	Session Error 1 Byte	Reconnects 2 Byte
Reserved 4 Bytes		
Uptime 4 Bytes		

8 24

Target Address

See request packet for details.

Action

The value of the request packet. Zero in messages.

Status

Values:

FAIL	0x00
SUCCESS	0x01

Error

Values:

ISCSI_MNI_ERR_INV_TARG	0x01
ISCSI_MNI_ERR_FULL	0x02
ISCSI_MNI_ERR_TARG_PRES	0x03
ISCSI_MNI_ERR_NOT_FOUND	0x04
ISCSI_MNI_ERR_INTERN	0x05

When an action request fails, this error field is set. On success or messages, this fild is zero.
ISCSI_MNI_ERR_INV_TARG - if the assigned target address is invalid (e.g. zero).
ISCSI_MNI_ERR_FULL - if the maxium of 64 iscsi targets is reached and a request packet with the ADD action was sent.
ISCSI_MNI_ERR_TARG_PRES - if a request packet if the ADD action was sent and the target address is already monitored.
ISCSI_MNI_ERR_NOT_FOUND - if a request packet if the DELETE action was sent and the target address is not currently monitored.

Target IP

The ip address of the iscsi target.

Target Index

The index of the iscsi target.

Starttime

The time when the monitoring process was started (in seconds since 2000).

Session Status

Values:

OFFLINE	0x00
ONLINE	0x01
ERROR	0x02

Session Error

Values:

ISCSI_ERR_CONNECT	0x31
ISCSI_ERR_LOGIN	0x34
ISCSI_ERR_INV_TARG_IDX	0x35
ISCSI_ERR_PWD	0x36
ISCSI_ERR_PROTO	0x37
ISCSI_ERR_TARG_NOT_REACH	0x38
ISCSI_ERR_NO_MEM	0x3a
ISCSI_ERR_SESS_CREATE	0x3b
ISCSI_ERR_INV_PARAMS	0x3c
ISCSI_ERR_SESS_NOT_FOUND	0x3d
ISCSI_ERR_DISCONN	0x3e
ISCSI_ERR_TIMEOUT	0x3f
ISCSI_ERR_SOCK	0x5f
ISCSI_SOCK_CLOSED	0x7f
ISCSI_ERR_TCP_CONN_RST	0x8f
ISCSI_ERR_IP_ZERO	0xa0

Reconnects

The number of times the session was reconnected since the begin of the monitoring process.

Uptime

The number of seconds the session is online.

Use this command to monitor iscsi targets.

If you write this command, all the targets of the request packets are added/delete to the current monitoring list.

If you read this command, you will get the description of all currently monitored iscsi targets in the response packet.

Every time the status of a monitored target changes, a message is sent out with the current status of this target.

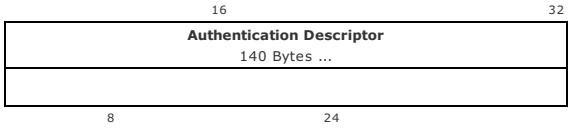
CONF_ISCSI_AUTH

Tag code	NumDes	Message	SNMP Support
0x0ab0	no	no	no

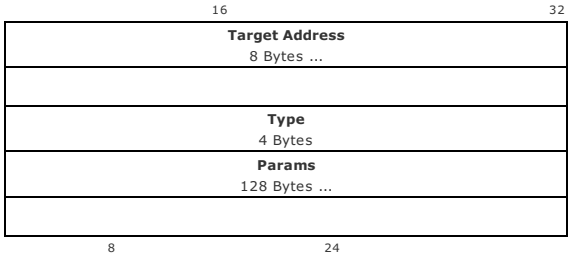
	Datatype	Access Level	Description
Read	p_octet	access_right_user	see detailed description
Write	p_octet	access_right_service	see detailed description

Payload Structure

Sequence of (BVIP enc/dec: 2, generic_dll: 8):

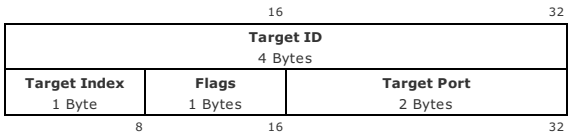


Authentication Descriptor



Target Address

The address of the iscsi target. If this field is set to zero, it is used as the default entry for authenticaion.



Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES). If this field is set to -1 (0xFFFFFFFF), this descriptor is used for all remaining targets, for that no explicit descriptor is provided (default).

Target Index

The target index of the iscsi target. If this field is set to -1 (0xFF), this descriptor is used for all remaining iscsi targets with the same ip, for that no explicit descriptor is provided (default).

Flags

Values:

http tunnel	0x01
not for target	0x02

Target Port

The port of the iscsi target. If this value is set to zero, the port configured in CONF_ISCSI_PORT is used.

NOTE

For an iscsi session, all authentication descriptors are searched for the target address. If this address is found, the authentication information is used.

If that descriptor is not found, but the array contains a descriptor with the same ip and the target index set to default (0xFF), the information in this record is used instead.

If that descriptor is not found, but the array contains an descriptor with the ip address set to default (0xFFFFFFFF), the information in this record is used instead.

If that descriptor is not found, no authentication will be performed for that iscsi session.

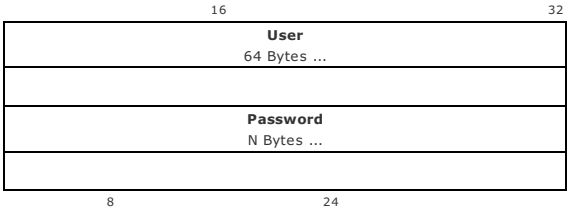
Type

Values:

NONE	0
CHAP	1
SMB	2

Params

(If Type != NONE)



User

The user name that is used for authentication.

Password

The password that is used for authentication (CHAP: N = 64 Bytes, SMB: N = 32 Bytes).

NOTE

On a read command, the characters of the password are replaced by the '*'. On a write command, the password is only stored, if not all characters equal the '*' sign. Otherwise the old stored value is retained.

CONF_ISCSI_SEG_SIZE

Tag code	NumDes	Message	SNMP Support
0x0aff	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Maximum segment size for ISCSI transfers (should be a 2^n value like 8192, 16384 or 65536)
Write	t_dword	access_right_service	Maximum segment size for ISCSI transfers (should be a 2^n value like 8192, 16384 or 65536)

CONF_ISCSI_DATARATE

Tag code	NumDes	Message	SNMP Support
0x0b00	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	KBit/s the ISCSI should transmit as maximum (0:=no limit send all data at once)
Write	t_dword	access_right_service	KBit/s the ISCSI should transmit as maximum (0:=no limit send all data at once)

CONF_ISCSI_LOWERDATARATE

Tag code	NumDes	Message	SNMP Support
0x0b47	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	KBit/s the ISCSI lower Limit for the Iscsi Data rate Throttling
Write	t_dword	access_right_service	KBit/s the ISCSI lower Limit for the Iscsi Data rate Throttling

CONF_ISCSI_READDATARATE

Tag code	NumDes	Message	SNMP Support
0x0b3a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	KBit/s the ISCSI should read as maximum (0:=no limit send all data at once)
Write	t_dword	access_right_service	KBit/s the ISCSI should read as maximum (0:=no limit send all data at once)

CONF_ISCSI_MULTIPATH

Tag code	NumDes	Message	SNMP Support
0x0bee	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the behaviour of iSCSI multipathing (0=off, 1=only use prefered pathes, 2=use all pathes, 3=use NetApp specific path selection)
Write	t_dword	access_right_service	set the behaviour of iSCSI multipathing (0=off, 1=only use prefered pathes, 2=use all pathes, 3=use NetApp specific path selection)

CONF_ISCSI_MULTIPATH_STATE

Tag code	NumDes	Message	SNMP Support
0x0c14	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Read the multipathing state of all active iSCSI connections
Write	p_octet	access_right_minimal	not supported

Payload Structure

The message normally contains one or more 0x00 tags or one 0x04 tag if there is no active iSCSI connection. Inside the 0x00 tag there should appear one 0x01, 0x05, 0x02, 0x03 and 0x06 tag.

16 32

Length 2 Byte	**First Group Tag (0x8000)** 2 Byte
Length 2 Byte	**First Subtag** 2 Byte
Payload (length - header) bytes	
Length 2 Byte	**Second Subtag** 2 Byte
Payload (length - header) bytes	
... Additional subtags (length - header) bytes	
Length 2 Byte	**Next Group Tag (0x8000)** 2 Byte
Length 2 Byte	**First Subtag** 2 Byte
Payload (length - header) bytes	
... Additional subtags (length - header) bytes	
... Additional group tags (length - header) bytes	

8 24

If there is no active iSCSI connection at the moment the command just returns the 'No active connection' tag without any payload.

16 32

Length 0x0004	**Tag** 0x0004

Length

Length of complete tag entry including length and tag field.

Values:

Length	0-65535

Tag

Tag describing the meaning of the following payload. See tag defintion below.

Values:

Group tag (Surrounds information for one connection)	0x8000
Main path IP	0x0001
Target index	0x0005
Connected IP	0x0002
Connection state	0x0003
Multipath support	0x0006
No active connection	0x0004

Payload

Binary payload. The meaning is defined trough the value of the tag field. The length is the amount of bytes given in the length field minus 4;

Values:

Binary data	0x..

Typical message

This is the payload when a target supports multipathing. The IP of the origin connection is 10.1.1.10. The target with index 2 on this IP is used. The connection runs over the IP 10.1.1.12 at the moment. This is a path which is marked as non optimized.

16 32

Length 0x001c	**Group Tag** 0x8000
Length 0x0008	**Subtag** 0x0001
IPv4 0x0a 0x01 0x01 0x0a	
Length 0x0008	**Subtag** 0x0005
Target Index 0x00 0x00 0x00 0x02	
Length 0x0008	**Subtag** 0x0002
IPv4 0x0a 0x01 0x01 0x0c	
Length 0x0008	**Subtag** 0x0003
Flags 0x00 0x00 0x1 0x00 (uint32 0x00000010)	
Length 0x0008	**Subtag** 0x0006
Multipath support 0x00 0x00 0x00 0x01 (uint32 0x00000001)	

8 24

When no iSCSI connection is requested/active at the moment (e.g. no recording running), the following payload is returned

16 32

Length 0x0004	**Tag** 0x0004

Tags

The command contains the following tags:

Tag 0x8000: Group tag

One group is marked by the 0x8000 starting tag. Each group gives information about one active iSCSI connection. It normally contains one 0x0001, one 0x0005, one 0x0002 and one 0x0003 tag to describe the multipath state of the connection.
The length field contains the length of all included subtags plus the group header itself.
The group tag is present to simplefy parsing and to join subtags to a unique connection description.

Tag 0x0001: Target IP

Original or main IP address of the iSCSI target. Through this IP the session was initialy created. If the length field is 8 the payload is a IPv4 address (network byte order). If the length field is 20 the payload contains IPv6.

Tag 0x0002: Connected IP

The IP address the connection uses at the moment. This may be the IP of an alternative path. If the length field is 8 the payload is a IPv4 address (network byte order). If the length field is 20 the payload contains IPv6.

Tag 0x0003: Connection status

The payload is 4 byte containing a 32 bit value (DWORD) in network byte order. The value must be interpreted as a bit field with the following meaning:

Bit 0 (0x00000001)	The connection is offline and not in a working state.
Bit 1 (0x00000002)	The connection is connected via the main IP (main path or prefered path) and working optimized.
Bit 2 (0x00000004)	The connection is connected via an alternative path. So The connection runs optimzied but not through the main path.
Bit 3 (0x00000008)	The connection runs through an alternative path which is not declared as being optimized or not optimized. Connection will go back to the main path when possible.
Bit 4 (0x00000010)	Connection runs on a non optimized path (lower performance) and will try to go back to an main path as soon as possible.
Bit 5 (0x00000020)	The session is about to be closed gracefully. There are no more users on this session and it will be closed.
Bit 6 (0x00000040)	The connection is connected via the main IP, but the regularly check of the availability of the alternative paths failed. So at lesast one alternative path might not be reachable.

Tag 0x0004: No active connection

If this tag is present as first tag, no other tags will be present in the message. It tells the receiver, that no active connection exists at the moment. So the iSCSI stack is idle.

Tag 0x0005: Target Index

The target index of the target which is addressed over the Target IP if multiple targets are available through this IP.

Tag 0x0006: Multipath support

If 0 the target has no multipath support and only one of the states Bit 0 (offline) and Bit 1 (connected optimized) is reported, as the target can only be on- or offline. It has no alternative pathes.
If this value is 1 the target supports multipathing and there are alternaitve pathes available. All states from above can be reported for the target.

CONF_ISCSI_FLUSH_DISVCACHE

Tag code	NumDes	Message	SNMP Support
0x0c19	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_service	flush the ISCSI discovery cache. See detailed description

Payload Structure

16	32
Length 2 Byte	**Tag (IP address)** 2 Byte
Payload 4 Bytes	
Additional tags (target index)... n Bytes	

8 24

Length

Length of the following tag including length and tag field.

Values:

Tag

Tag which describes the meaning of the following field. This command supports both variants of passing IP adresses. You can provide an IP with tag 0x01 either an IPv4 or IPv6. Then the length field indicates which IP version is included (4 = IPv4, 16 = IPv6). To pass IPv6 you can also use tag 0x02.

Values:

IPv4 or IPv6 Address. Indicated by the length field.	0x01
IPv6 Address	0x02
Target Index	0x03

Payload

Payload of the tag (4 bytes in case of IPv4, 16 bytes in case of IPv6). The target index has 4 byte payload and contains the target index in network byte order.

Values:

General Description

This command flushes one entry in iSCSI discovery cache. Providing an IP and target index is mandatory.

Tag 0x01: IPv4

The payload contains an IPv4 address in network byte order which is to be flushed in the discovery cache. The payload must be 4 bytes.

Tag 0x02: IPv6

The payload contains an IPv6 address in network byte order which is to be flushed in the discovery cache. The payload must be 16 bytes.

Tag 0x03: Target Index

The payload of this tag contains a target index. If there are multiple targets on one IP you can flush the cache for individual target entries by passing the target index in network byte order.

CONF_ISCSI_DISV_CACHE

Tag code	NumDes	Message	SNMP Support
0x0c30	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Read out discovery cache entries
Write	p_octet	access_right_service	Write one discovery cache entry. ##desc:.\doc\CONF_ISCSI_DISV_CACHE.htm##

Payload Structure

This command can be used to read and write individual entries of the discovery cache.
The command supports reading and writing cache entries.

Read direction: Payload Structure

On input you can provide an IP address and/or a LUN number structured in the following tagged payload. IP addresses (tag 0x0001) can have the length 0x0008 (IPv4) or 0x0014 (IPv6).
The Target Index tag (0x000b) must have the length 0x0008 and must contain a target index in network byte order
The LUN tag (0x0002) must have the length 0x0008 and must contain a 32bit LUN number in network order.
The payload can contain all three tags. With it's combination you can control the output of the command.

- If provide no parameters (tags) the command returns all known IP of all target indexes and LUNs in the discovery cache. Be aware, that the output could get too long for the output buffer and there might be missing entries
- If you only provide an IP address, the command will return all targets and all known LUNs and the paths for one IP in the discovery cache.
- If you provide both IP address and a target index, the command will return all known LUNs and the paths for one target in the discovery cache.
- If you only provide a LUN number, the paths for this LUN in all discovery cache entries will be returned. There is a reserved LUN number 0xFFFFFFFF which will always return only the first LUN of an entry in the discovery cache. If you provide this reserved LUN number without an IP address, you will get all entries in the discovery cache with only the paths for the first LUN. This is a good way to get the IP and target index of all known targets in the discovery cache without overflowing the output buffer.
- If you provide an IP address, a target index and a LUN number the command will return only the paths for this target and the given LUN.

The answer is structured as described in Cache Entry encoding.
If there is no cache entry available which matches the given filter, the command returns the 'Tag 0x0003: No entry available' tag. In case of this command the tag always return 0x00000000 as error code (no error).

16	32
Length 2 Byte	**Tag (0x0001, 0x0002 or 0x000b)** 2 Byte
Payload (length - header) bytes	
Additional tags... n bytes	

Write direction: Payload Structure

In write direction the command expects the Cache Entry encoding described below. This encoded cache entry is written to the discovery cache. Cache entries which are written with this command are stored in persistent cache and are persistent over a device reset.
In write direction this command must carry a valid external instance bitmask which has to be unequal 0. If it is 0 the command is returned with an error.
To delete such an entry you must provide the Cache Entry encoding with only the target IP, target index and external instance tag. The entry is cleared, when all external instance bits are cleared.

Cache Entry encoding (answer payload for read, input payload for write)

Every cache entry is started with a surrounding start tag (0x8009) followed by the IP for which this entry was generated (tag 0x000a). After this IP entry the target index is indicated to resolve a unique target on this IP (tag 0x000b). A surrounding tag (0x8008) marks one LUN entry which has several path entries, each surrounded by a 0x8004 tag.
If there is no entry the 'Tag 0x0003: No entry available' tag is returned.

16	32
Length 2 Byte	**Cache entry Tag (0x8009)** 2 Byte
Length 2 Byte	**Target IP tag (0x000a)** 2 Byte
Payload (length - header) bytes	
Length 2 Byte	**Target Index tag (0x000b)** 2 Byte
Payload (length - header) bytes	
Length 2 Byte	**IQN string tag (0x000c)** 2 Byte
IQN string (length - header) bytes	
	external instance bitfield tag

Length 2 Byte	**(0x000d)** 2 Byte
External instance bitfield (32bit) (length - header) bytes	
Length 2 Byte	**LUN Entry Group Tag (0x8008)** 2 Byte
Length 2 Byte	**Path Entry Group Tag (0x8004)** 2 Byte
Length 2 Byte	**Path IP Tag (0x0005)** 2 Byte
IP of the path (16 or 4 byte length) (length - header) bytes	
Length 2 Byte	**LUN Tag (0x0006)** 2 Byte
LUN number on this path (length - header) bytes	
Length 2 Byte	**Flags Tag (0x0007)** 2 Byte
Flags see below (length - header) bytes	
... Additional Path Entry Group Tag (0x8004) (length - header) bytes	
... Additional LUN Entry Group Tag (0x8008) (length - header) bytes	
... Additional Cache entry Tag (0x0009) (length - header) bytes	

In case of an error or there is no matching cache entry the following payload is returned:

16	32
Length 0x0008	**Tag** 0x0003
Error code (4bytes network order)	

Length

Length of complete tag entry including length and tag field

Values:

Length	0-65535

Tag

Tag describes the meaning of the following payload. See tag definition below.

Values:

Discovery cache entry tag	0x8009
Target IP	0x000a
Target Index	0x000b
IQN string	0x000c
External instance	0x000d
LUN entry tag	0x8008
Path entry	0x8004
Path IP	0x0005
LUN	0x0006
Flags	0x0007

Typical message

This is the payload for a command which addresses (10.1.1.10), target index 3 and one indivdual LUN (4).

Input data:

16	32
Length 0x0008	**IP tag** 0x0001
IPv4 0x0a 0x01 0x01 0x0c	
Length 0x0008	**Target Index** 0x000b
Target Index 0x00 0x00 0x00 0x03	
Length 0x0008	**Subtag** 0x0002
LUN 0x00 0x00 0x00 0x04	

Output data

16	32
Length 0x0084	**Cache entry** 0x8009
Length 0x0008	**Target IP** 0x000a
IPv4 0x0a 0x01 0x01 0x0a	
Length	**Target Index**

0x8008	0x000b
Number 0x00 0x00 0x00 0x03	
Length 0x0012	**IQN String** 0x000b
IQN String "iqn-bosch-1234"	
Length 0x0008	**External instance** 0x000d
External instance 0x00 0x00 0x00 0x01	
Length 0x0074	**LUN entry Tag** 0x8008
Length 0x001c	**Path entry tag** 0x8004
Length 0x0008	**IP** 0x0005
IPv4 0x0a 0x01 0x01 0x0c	
Length 0x0008	**LUN** 0x0006
LUN number 0x00 0x00 0x00 0x04	
Length 0x0008	**Flags** 0x0007
Flags bitfield 0x00 0x00 0x00 0x07	
More Paths (0x8004) ...	

When no cache entry is available, the following payload is returned:

16	32
Length 0x0004	**Tag** 0x0003

Tags

The command contains the following tags:

Tag 0x8009: Cache entry tag

Each cache entry start with this tag and is followed by an target IP tag. If there is no multipathing information, there will be no following 'LUN entry tags'.

Tag 0x000a: Target IP

Original or main IP address of the iSCSI target. Through this IP the session was initially created. If the length field is 8 the payload is a IPv4 address (network byte order). If the length field is 20 the payload contains IPv6.

Tag 0x000b: Target Index

Target index on the main IP. This is the index of the target if there are multiple targets available on one IP.

Tag 0x000c: IQN string

IQN string of the target. This is the unique name identifier the target returns at discovery and which is needed at login state to address the target. In multipathing this IQN is used for all connection IP addresses.

Tag 0x000d: External instance

The external instance is a 32 bit wide bitfield. It is used to identify who has set this cache entry. In VRM environment the first VRM may set bit 0. The secondary VRM will use bit 1. With this bitfield it is tracked, who wrote this cache entry. If one instance deletes the cache entry, only the bit used by this instance is cleared. If any other bits are still set, the entry is not deleted. When this bitfield becomes zero at a delete request, the entry is really deleted. **ATTENTION:** This field must always have a value other then 0. Writing a cache entry from outside world with external instance equal 0 will cause a write error.

Tag 0x8008: LUN entry

Surrounding tag for one LUN entry. This tag will have several 'Path entry' tags as subtags.

Tag 0x8004: Path entry

Tag which marks one path entry. The first path entry will always be the path on the main path. This entry will have the following subtags:

Tag 0x0005: Path IP

IP address of this path. If the length field is 8 the payload is a IPv4 address (network byte order). If the length field is 20 the payload contains IPv6.

Tag 0x0006: LUN number

The number of the LUN the LUN will have on this path. The first entry in the paths list marks the LUN number on the main path.

Tag 0x0007: Flags

Conditions for this path

Bit 0 (0x00000001)	Target supports TPGS (Target Portal Group Support) on this path. Necessary for proper multipathing support.
Bit 1 (0x00000002)	This path is a prefered path
Bit 2 (0x00000004)	This path is an active path (but maybe not preferred if bit 1 is cleared).
Bit 3 (0x00000008)	This path is not optimal and should only be used as backup path in case of all preferred pathes fail.

Tag 0x0003: No entry available

If this tag is present as first tag, no other tags will be present in the message. It tells the receiver, that there are no matching discovery cache entries.
The tag is followed by an error code (4 byte in network order). If the error code is 0x00000000 (ISCSI_ERR_NONE) there was simply no existing cache entry available. For CONF_ISCSI_DISV_CACH command the error code is always 0x00000000. For CONF_ISCSI_MP_DISCOVERY any of the defined iSCSI error codes can occur.

ISCSI_ERR_NONE	0x00
ISCSI_ERR_CONNECT	0x31
ISCSI_ERR_INV_LUN	0x33
ISCSI_ERR_LOGIN	0x34
ISCSI_ERR_INV_TARG_IDX	0x35
ISCSI_ERR_PWD	0x36
ISCSI_ERR_PROTO	0x37
ISCSI_ERR_TARG_NOT_REACH	0x38
ISCSI_ERR_NO_MEM	0x3a
ISCSI_ERR_SESS_CREATE	0x3b
ISCSI_ERR_INV_PARAMS	0x3c
ISCSI_ERR_SESS_NOT_FOUND	0x3d
ISCSI_ERR_DISCONN	0x3e
ISCSI_ERR_TIMEOUT	0x3f
ISCSI_ERR_TARGET_NOT_SUPP	0x40
ISCSI_ERR_TARGET_SESSION_LIMIT	0x41
ISCSI_ERR_CMD_NOT_SUPP	0x42
ISCSI_ERR_TARGET_NOT_FOUND	0x43
ISCSI_ERR_SOCK	0x5f
ISCSI_ERR_TARG_PM	0x6f
ISCSI_SOCK_CLOSED	0x7f
ISCSI_ERR_TCP_CONN_RST	0x8f
ISCSI_ERR_INTR_NOT_SUPP	0x9f
ISCSI_ERR_IP_ZERO	0xa0
ISCSI_ERR_OUT_OF_RES	0xa1

CONF_ISCSI_MP_DISCOVER

Tag code	NumDes	Message	SNMP Support
0x0c32	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Do a multipath discover
Write	p_octet	access_right_service	not supported

Payload Structure

The payload must contain the tag 0x0001 **and** the tag 0x0002. Optionally you can pass one or more tag 0x0003 to tell the command to only scan the given pathes. The payload of the tag 0x0001 and tag 0x0003 can either be IPv4 or IPv6. Which IP is contained is indiacted by the tag length. 0x0008 is IPv4 0x0014 is IPv6.
The tag 0x0002 indicates a target index.

16	32
Length 2 Byte	**Tag (0x0001)** 2 Byte
IPv4 or IPv6 4 or 16 Bytes	
Length 2 Byte	**Tag (0x0002)** 2 Byte
Target Index 4 Bytes	
OPTIONAL 1 or more 0x0003 tag indicating alternative pathes to scan.	
Length 2 Byte	**Tag (0x0003)** 2 Byte
IPv4 or IPv6 4 or 16 Bytes	

Length

Length of the following tag including length and tag field.

Values:

Tag

Tag which describes the meaning of the following field

Values:

IP Address	0x0001
Target Index	0x0002
Alternative path	0x0003

Payload

Payload of the tag 0x0001 (4 bytes in case of IPv4, 16 bytes in case of IPv6) indicates the main path to the target to scan.

Payload of the tag 0x0002 is the target index (4 byte number) on network byte order

Payload of each tag 0x0003 (4 bytes in case of IPv4, 16 bytes in case of IPv6) gives the IP of an alternative path to scan. If not given all found pathes are scanned.

General Description

This command initiates a multipath discovery of a target. This means iSCSI stack tries to detect all known alternative pathes for all LUNs for the given main path.
The user can select alternative pathes to be scanned with the 0x0003 tag. If no 0x0003 tag is given all pathes found at discovery are scanned.
ATTENTION: The command can take up to the amount of scanned pathes multiplied with 10 seconds to complete! This scan does not create any persistent discovery cache entries. All cached data is cleared after the scan.

Response

Response payload

The payload of the response is an encoded discovery cache entry which show the scan result. The format is the same is described in the CONF_ISCSI_GET_DISV_CACHE command.

iSCSI specific errors

The following specific erros might be returned on RCP level. If such an error occurs the command was not executed.
ISCSI_ERR_INV_PARAMS (0x3c) - Error parsing the command (error in tag structure)
ISCSI_ERR_TARGET_SESSION_LIMIT (0x41) - Too many scan active (only one to the same IP is allowed). Retry later.
ISCSI_ERR_NO_MEM (0x3a) - No free job to do the scan. Retry later.

The command can also return the 'No cache entry' tag (0x0003) followed by an error code. With this the command returns errors which occur while discovering the target. All known iSCSI error codes can occur there (see CONF_DISV_CACHE for a list of error codes).

CONF_HD_MAX_NUMBER_OF_PARTITIONS

Tag code	NumDes	Message	SNMP Support
0x0900		no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete
Write	void	access_right_user	not supported

CONF_HD_MAX_SLICES_PER_TRACK

Tag code	NumDes	Message	SNMP Support
0x090a	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete
Write	void	access_right_user	not supported

CONF_HD_MAX_ALARM_TRACKS_PER_PARTITION

Tag code	NumDes	Message	SNMP Support
0x0919	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete
Write	void	access_right_user	not supported

CONF_HD_PARTITIONS_RECORDING

Tag code	NumDes	Message	SNMP Support
0x091a	yes (1 - primary recording ,2 - secondary recording)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	list of cams with running recording
Write	void	access_right_user	not supported

CONF_HD_PARTITION_RECORDING

Tag code	NumDes	Message	SNMP Support
0x0a03	yes (cam)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	recording status of cam for primary recording (1 = running, 0 = not running)
Write	t_dword	access_right_user	not supported

CONF_HD_PARTITION_RECORDING_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4d	yes (cam)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	recording status of this cam for secondary recording (1 = running, 0 = not running)
Write	t_dword	access_right_user	not supported

CONF_HD_PARTITION_FILE_INFO

Tag code	NumDes	Message	SNMP Support
0x0901	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns a list of files of a replay session (session id needed). This command works for local replay only. See detailed description
Write	void	access_right_user	not supported

This command initiates a search to get the times where data is available within a recording.

The Session ID paramter must be set (a connect primitive must have been preceded). Recording mode can be time recording (1) or alarm recording (2 and 3), only for command CONF_SPAN_PARTITION_FILE_INFO there are seperagte files for pre (2) and post (3) alarm recording.

Search is performed from a start point on the timeline towards an end point. If the end point is prior to the start point search is performed backwards.

To get all available results of an recording a search from 0 to 0xFFFFFFFF needs to be done.

The number of results is limited to 256 per response. The result are always in ascending order independent if the search is performed forward or backward. The File ID always increases on span recording regions if new files will be created.

To get all results the command has to be called multiple times.

In that case the start or stop times needs to be updated with every call: If a forward search is executed the start time of the next request must be the stop time of the last retrieved result. If a bachward search is executed the start time of the next request must be the start time of the first retrieved result.

A search is completed if the number of returned results is smaller than 256 or if the endpoint of the search is greater or equal than than the endpoint of the last result. In case of forward search a virtual start point (equal to the start point of the search) will be inserted if the search start point is range with recording.

Some examples: Forward search (data is available from time A to time B):

1. A search is executed from time 0 to time B. In that case the reponse is: secStart=A, secStop=B
2. A search is executed from time 0 to time 0xFFFFFFFF. In that case the reponse is: secStart=A, secStop=B
3. A search is executed from time C (C > A and C < B) to time B. In that case the reponse is: secStart=C, secStop=B
4. A search is executed from time A to time C (C > A and C < B). In that case the reponse is: secStart=A, secStop=C

Backward search (data is available from time A to time B):

1. A search is executed from time 0xFFFFFFFF to time A. In that case the reponse is: secStart=A, secStop=B
2. A search is executed from time 0xFFFFFFFF to time 0. In that case the reponse is: secStart=A, secStop=B
3. A search is executed from time C (C < B and C > A) to time A. In that case the reponse is: secStart=A, secStop=C
4. A search is executed from time B to time C (C < B and C > A). In that case the reponse is: secStart=C, secStop=B

Request

Payload Structure

16	32
Search Start Time 4 Bytes	
Search Stop Time 4 Bytes	
maxEntries 4 Bytes	
flags 4 Bytes	
8	24

Search Start Time

Seconds since 2000

Search Stop Time

Seconds since 2000

maxEntries

Max Number of entries, limited to 256

Res

1. YOUNGER_OR_EQUAL_FLAG: must be set in case of forward search

2. UTC_FLAG: start and stop time are utc times (request)

Reply

Payload Structure (sequence of)

16	32
Start Time 4 Bytes	
Stop Time 4 Bytes	
Flags 32 Bits	
File ID 4 Bytes	
8	24

Start Time

Seconds since 2000.

Stop Time

Seconds since 2000.

Flags

Values:

Bit 0	Recording Running (actual recording is running on this file or recording not closed regulary)
Bit 1	Recording Overwriting (recording takes place in a ring and old recording data will be overwritten)
Bit 2	Alarm Input (there are input alarms in this file)
Bit 3	Alarm Motion (there are motion alarms in this file)
Bit 4	New Alarm (obsolete)
Bit 5	Video Loss (there are video loss in this file)
Bit 6 - 7	Recording mode: 1 - time recording, 2 - alarm recording (pre alarm), 3 - alarm recording (post alarm) value 2 and 3 only in CONF_SPAN_PARTITION_FILE_INFO distinguishable, for CONF_PARTITION_FILE_INFO these two values have the meaning of a full alarm rec file
Bit 8-15	Track Fill Level (fill level in percent, always 100 % on filled ring recording)
Bit 16	Alarm Remote (there are virtual/remote alarms in this file, see CONF_HD_MGR_SIGNAL_ALARM)
Bit 17	Audio (there are audio data in this file)
Bit 18	Meta (there are meta data in this file)
Bit 19-20	Reserved
Bit 21	Offline (VRM only)
Bit 22	Protected (VRM only)
Bit 23-28	Time Zone (Quarter hours)
Bit 29	Time Zone Sign
Bit 30-31	Reserved

CONF_HD_PARTITION_PROP

Tag code	NumDes	Message	SNMP Support
0x090b	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete
Write	p_octet	access_right_service	(obsolete) see detailed description (video type is obsolete, depends on CONF_CODER_VIDEO_TYPE_RESTRICTION settings)

Payload Structure

Number 4 Bytes		
Char 1 1 Byte	62 x 1 Byte	**Char 64** 1 Byte

Total Size 4 Bytes

Reserved 4 Bytes

Video Type 1 Byte	**Reserved** 1 Byte	**Storage Type** 1 Byte	**Enc. Preset** 1 Byte

Number Of Alarm Tracks (obsolete) 4 Bytes

Size Of Alarm Track (obsolete) 4 Bytes

Video input channel 4 Bytes

Encoder index 1 Byte	**Reserved** 3 Bytes

Reserved 32 Bytes ...

Char 1 - 64

Name of the partition (obsolete).

Size (obsolete)

In Megabyte

Video Type (obsolete, depends on CONF_CODER_VIDEO_TYPE_RESTRICTION settings)

Values:

No	0x00
Mpeg2	0x01
Mpeg4	0x03
H264	0x04

Storage Type (obsolete)

Values:

Linear	0x00
Ring	0x01

Enc. Preset

Number of the encoder preset 1 - 8. (obsolete here, can bet set in the recording profiles)

Size Of Alarm Track (obsolete)

In Megabytes.

Video input channel

Bitfield of used video channels (Bit0=channel 1...).

Encoder index

Number of the encoder instance. (obsolete here, can bet set in the recording profiles)

CONF_HD_PARTITION_PROP for Span Recording

This command cannot be set while the recording is running.

CONF_HD_PARTITION_PROP_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4e	yes (cam)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete, payload as in CONF_HD_PARTITION_PROP
Write	p_octet	access_right_service	(obsolete)partition properties for secondary recording, payload is the same as the command CONF_HD_PARTITION_PROP

CONF_HD_SIZE_MB

Tag code	NumDes	Message	SNMP Support
0x090c	yes (0: default pm, >0: storage medium type)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	read the total size of a local storage in megabytes
Write	-	access_right_service	not writable

CONF_HD_PARTITION_GEO

Tag code	NumDes	Message	SNMP Support
0x090d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete
Write	p_octet	access_right_service	obsolete

CONF_HD_RECORD_SCHEDULE

Tag code	NumDes	Message	SNMP Support
0x0a0b	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read primary recording schedule, see detailed description
Write	p_octet	access_right_service	set primary recording schedule(effect takes place imidiately), see detailed description

Payload Structure

Sat. 1 4 Bits	94 x 4 Bits	**Sat. 96** 4 Bits	**Sun. 1** 4 Bits	94 x 4 Bits	**Sun. 96** 4 Bits
Mon. 1 4 Bits	94 x 4 Bits	**Mon. 96** 4 Bits	**Tue. 1** 4 Bits	94 x 4 Bits	**Tue. 96** 4 Bits
Wed. 1 4 Bits	94 x 4 Bits	**Wed. 96** 4 Bits	**Thu. 1** 4 Bits	94 x 4 Bits	**Thu. 96** 4 Bits
Fri. 1 4 Bits	94 x 4 Bits	**Fri. 96** 4 Bits			

Sat. 1 - 96

96 entries of recording profile numbers for Saturday, each represents the recording profile for a 15 min time period. First entry is from 00:00 to 00:15. The following entries are for the following 15 min time periods until 24:00.

Values:

recording off	0
recording profile numbers	1-10

Sun. 1 - 96

96 entries of recording profile numbers for Sun.

Mon. 1 - 96

96 entries of recording profile numbers for Mon.

Tue. 1 - 96

96 entries of recording profile numbers for Tue.

Wed. 1 - 96

96 entries of recording profile numbers for Wed.

Thu. 1 - 96

96 entries of recording profile numbers for Thu.

Fri. 1 - 96

96 entries of recording profile numbers for Fri.

HD_RECORD_SCHEDULE

This command reads/writes the recording schedule for all cameras (primary and secondary recording). The entries are 4 bit each and represent a recording profile for a 15 min time period(payload 336 bytes total). The recording profiles with same profile number can be different for each camera. See command HD_RECORD_PROFILES. The schedule will be written to the storage medium. In case of span recording mode (see CONF_RECORD_MODE_SPANS) the schedule will be only read from or stored in the config. This schedule is used as default schedule for each span that is mounted by this device for recording.

CONF_HD_RECORD_SCHEDULE_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a49	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete, only on global schedule for primary and secondary recording
Write	p_octet	access_right_service	obsolete, only on global schedule for primary and secondary recording on all cams

CONF_HD_RECORD_HOLIDAYS

Tag code	NumDes	Message	SNMP Support
0x0a0c	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read holiday schedule list for primary recording, see detailed description
Write	p_octet	access_right_service	write holiday schedule list (effect takes place imidately), see detailed description

Payload Structure

holiday schedule entry 1 52 Bytes ...

N x 52 Bytes	

holiday schedule entry N 52 Bytes ...

holiday schedule entry 1 - N

schedule entry specifing a holiday and it's recording schedule. max. 25 entries.

CONF_HD_RECORD_HOLIDAYS

This command reads/writes the Holiday table for the primary recording. On holidays the recording uses the special schedule instead of the standard schedule. The table contains N entries (max. 25) each entry specifies one or more holidays in a row followd by a recording schedule. the payload sends only valid holiday entries, so the size of N depends on the number of valid entries in the table.

Holiday Schedule Entry Structure

day	month	year	number of days
1 Byte	1 Byte	1 Byte	1 Byte

reco 1		reco 96
4 Bits	94 x 4 Bits	4 Bits

day

day of the month, which is a holiday

Values:

invalid	0
day of the month	1 - 31

month

month of the year

Values:

invalid	0
jan	1
feb to nov	2 - 11
dec	12

year

year since 2000

Values:

year 2000 to 2255	0 - 255

number of days

number of days that follow the specified holiday, which are also scheduled as holidays. If only the specified date is the holiday, this field should be set to 1.

Values:

invalid	0
number of days	1 - 255

record schedule 1 - 96

this is the recocording schedule containing 96 entries of recording profile numbers for the holiday, each represents the recording profile for a 15 min time period. First entry is from 00:00

to 00:15. The following entries are for the following 15 min time periods until 24:00.

Values:

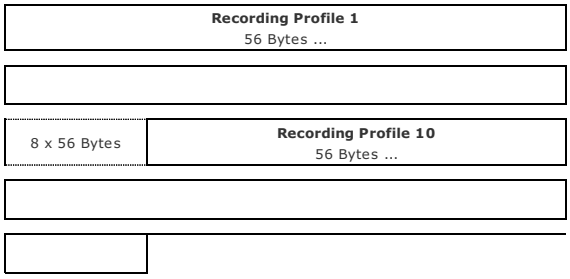
recording off	0
recording profile numbers	1 -10

CONF_HD_RECORD_PROFILES

Tag code	NumDes	Message	SNMP Support
0x0a0d	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read recording profiles, see detailed description
Write	p_octet	access_right_service	set recording profiles (effect takes place imidiatly), see detailed description

Payload Structure



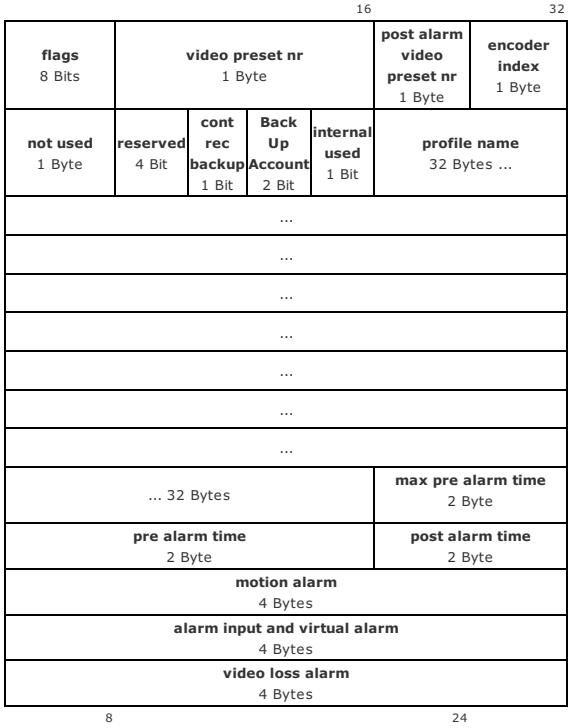
Recording Profile 1 - 10

Upto 10 recording profiles (at least 1) for the camera specified by the num parameter, see detailed description for a recording profile

HD_RECORD_PROFILES

Read/Write upto ten recording profiles (at least one) for a camera, payload upto 560 bytes total. The profiles will be written to the device config. These profiles are used as default profiles for each span that is mounted by this cam for span recording. If less than 10 profiles will be written, the remaining profiles stay unchanged.

recording profile structure



flags

Flags:

ALARM_RECORDING_PRE_ALARM_RING	0x01	flag for alarm recording, if set alarm recording on the alarm tracks will be active and pre alarm will be recorded into a ring on the storage
AUDIO_RECORDING_DISABLE	0x02	flag for audio recording, if set audio will be disabled for recording
		flag for meta recording, if set

META_RECORDING_DISABLE	0x04	meta will be disabled for recording
ALARM_FILE_BACKUP	0x08	flag for alarm file backup, if set alarm file back up will be active
ALARM_RECORDING_PRE_ALARM_BUFFER	0x10	flag for alarm recording, if set alarm recording on the alarm tracks will be active and pre alarm will be recorded into the recording buffer and will be flushed on alarm
ALARM_RECORDING_PRE_ALARM_AUTO	0x11	both flags for alarm recording, first the prealarm recordes into the recording buffer, if the prealarm time is to high for the buffer size, it starts recording into a ring on storage
IMMEDIATE_ALARM_BACKUP	0x20	immediate backup from record buffer
MANAGED_BY_ONVIF	0x40	a onvif video encoder configuration is active

video preset nr

video preset number used for recording

Values:

no recording on this profile	0
video preset nr	1 - 8

post alarm video preset nr

video preset number used for recording after occurence of an alarm

Values:

not set, uses video preset nr if set	0
video preset nr for post alarm	1 - 8

encoder index

index of the encoder stream starting with 0, index 255 is for the backup output

cont rec backup

only for continuous buffered recording, if set the records will be backuped to the recording, its an alternativ way of configuration to using a backup account of typ record, that way no account is wasted in the configuration

back up account

Backup account Number from 0 (first account) to 3(fourth account), for choosing the back up account in case activated ALARM_FILE_BACKUP flag

profile name

profile name is a zero terminated string

max pre alarm time

only relevant for backup recording, it needs to know the max pre alarm time, in other cases it should be set to zero

pre alarm time

only relevant for alarm recording, if set, this time in seconds is the time the recording will be stored up to the alarm event

post alarm time

if set, this time in seconds is the time after an alarm event for that the post alarm video preset is used if set. After this time elapses, the recording will return to the standard video preset nr, if alarm recording, the recording will also stop the recording on the actual track and start a new one

motion alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit 31 <-> Alarm Nbr. 32

alarm input and remote alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit n <-> Alarm Nbr. n for alarm inputs (n = number of input alarms), Bit (32 - m) <-> Virt. Alarm Nbr. m; ... ... ; Bit 30 <-> Virt. Alarm Nbr. 2; Bit 31 <-> Virt. Alarm Nbr. 1 (reverse order) for virtual alarms caused by rcp command CONF_VIRTUAL_ALARM_STATE (m = number of virt. alarms)

video loss alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit 31 <-> Alarm Nbr. 32

CONF_HD_RECORD_PROFILES_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a91	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read recording profiles for secondary recording, payload is the same as in CONF_HD_RECORD_PROFILES
Write	p_octet	access_right_service	set recording profiles for secondary recording (effect takes place imidiatly) , payload is the same as in CONF_HD_RECORD_PROFILES

CONF_HD_RECORD_PROFILES_V2

Tag code	NumDes	Message	SNMP Support
0x0ad0	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read recording profiles
Write	p_octet	access_right_service	write recording profiles

Payload Structure

Recording Profile 1 56 Bytes ...	
8 x 56 Bytes	**Recording Profile 10** 56 Bytes ...

Recording Profile 1 - 10

Upto 10 recording profiles (at least 1) for the camera specified by the num parameter, see detailed description for a recording profile

HD_RECORD_PROFILES_V2

Read/Write the ten recording profiles for a camera, payload upto 560 bytes total. The profiles will be written to the device configuration. These profiles are used as default profiles for each span that is mounted by this cam for span recording. If less than 10 profiles will be written, the remaining profiles stay unchanged.

recording profile structure

16 32

flags 8 Bits	**stream config nr** 1 Byte				**post alarm stream config nr** 1 Byte	**encoder index** 1 Byte
post alarm profile 1 Byte	**reserved** 4 Bit	**cont rec backup** 1 Bit	**Back Up Account** 2 Bit	**internal used** 1 Bit	**profile name** 32 Bytes ...	
...						
...						
...						
...						
...						
...						
...						
... 32 Bytes					**max pre alarm time** 2 Byte	
pre alarm time 2 Byte					**post alarm time** 2 Byte	
motion alarm 4 Bytes						
alarm input and remote alarm 4 Bytes						
video loss alarm 4 Bytes						

8 24

flags

Flags:

ALARM_RECORDING_PRE_ALARM_RING	0x01	flag for alarm recording, if set alarm recording on the alarm tracks will be active and pre alarm will be recorded into a ring on the storage
AUDIO_RECORDING_DISABLE	0x02	flag for audio recording, if set audio will be disabled for recording
META_RECORDING_DISABLE	0x04	flag for meta recording, if set meta will be disabled for recording

ALARM_FILE_BACKUP	0x08	flag for alarm file backup, if set alarm file back up will be active
ALARM_RECORDING_PRE_ALARM_BUFFER	0x10	flag for alarm recording, if set alarm recording on the alarm tracks will be active and pre alarm will be recorded into the recording buffer and will be flushed on alarm
ALARM_RECORDING_PRE_ALARM_AUTO	0x11	both flags for alarm recording, first the prealarm recordes into the recording buffer, if the prealarm time is to high for the buffer size, it starts recording into a ring on storage
IMMEDIATE_ALARM_BACKUP	0x20	immediate backup from record buffer

stream config nr

stream configuration number used for recording

Values:

no recording on this profile	0
stream configuration nr for pre alarm	1 - 2
I-frames only from stream 1 for pre alarm	3

post alarm stream config nr

stream configuration used for recording after occurence of an alarm

Values:

not set, uses stream config nr if set	0
stream configuraton nr for post alarm	1 - 2
I-frames only from stream 1 for post alarm	3

encoder index

index of the encoder stream starting with 0, index 255 is for the backup output

post alarm profile

encoder profile for postalarm, only some values of the profile will be used to modify the uses encoder profile for the post alarm stream

Values:

not set, no modification of the actual enc profile	0
encoder profile nr for post alarm configuration	1 - 8

cont rec backup

only for continuous buffered recording, if set the records will be backuped to the recording, its an alternativ way of configuration to using a backup account of typ record, that way no account is wasted in the configuration

back up account

Backup account Number from 0 (first account) to 3(fourth account), for choosing the back up account in case activated ALARM_FILE_BACKUP flag

profile name

profile name is a zero terminated string

max pre alarm time

only relevant for backup recording, it needs to know the max pre alarm time, in other cases it should be set to zero

pre alarm time

only relevant for alarm recording, if set, this time in seconds is the time the recording will be stored up to the alarm event

post alarm time

if set, this time in seconds is the time after an alarm event for that the post alarm video preset is used if set. After this time elapses, the recording will return to the standard video preset nr, if alarm recording, the recording will also stop the recording on the actual track and start a new one

motion alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit 31 <-> Alarm Nbr. 32

alarm input and remote alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit n <-> Alarm Nbr. n for alarm inputs (n = number of input alarms), Bit (32 - m) <-> Virt. Alarm Nbr. m; ... ... ; Bit 30 <-> Virt. Alarm Nbr. 2; Bit 31 <-> Virt. Alarm Nbr. 1 (reverse order) for virtual alarms caused by rcp command CONF_VIRTUAL_ALARM_STATE (m = number of virt. alarms)

video loss alarm

The bits represents the activation. Bit 0 <-> Alarm Nbr. 1; Bit 1 <-> Alarm Nbr. 2; ... ... ; Bit 31 <-> Alarm Nbr. 32

CONF_HD_RECORD_PROFILES_V2_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0ad1	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get recording profiles for secondary recording, payload is the same as in CONF_HD_RECORD_PROFILES_V2
Write	p_octet	access_right_service	set recording profiles for secondary recording, payload is the same as in CONF_HD_RECORD_PROFILES_V2

CONF_HD_ALARM_MOTION

Tag code	NumDes	Message	SNMP Support
0x0916	yes (partition)	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	alarm motion - obsolete
Write	flag	access_right_service	write alarm motion - obsolete

CONF_HD_ALARM_INPUT

Tag code	NumDes	Message	SNMP Support
0x0917	yes (partition)	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	alarm input - obsolete
Write	flag	access_right_service	write alarm input - obsolete

CONF_HD_MGR_START

Tag code	NumDes	Message	SNMP Support
0x0913	yes (cam)	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	return TRUE when manager is on or when primary span recording is on
Write	flag	access_right_service	start/stop the recording manager and set config to recording on:1/off:0. Can cause recording if time/alarm recording selected.

CONF_HD_MGR_START_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a46	yes (cam)	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	return TRUE when secondary span recording is on
Write	flag	access_right_service	start/stop the recording manager and set config to recording on:1/off:0. Can cause recording if time/alarm recording selected.

CONF_HD_MGR_STOP

Tag code	NumDes	Message	SNMP Support
0x0914	cam	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	return TRUE when manager is off or when primary span recording is off
Write	flag	access_right_service	stop/start the recording manager and set config to primary recording off:1/on:0.

CONF_HD_MGR_STOP_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a47	partition or cam	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	return TRUE when secondary span recording is off
Write	flag	access_right_service	stop/start the recording manager and set config to secondary recording off:1/on:0.

CONF_HD_MPEG4_ACTIVE

Tag code	NumDes	Message	SNMP Support
0x093c	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	(obsolete) signals an active hdd MPEG4 partition
Write	-	access_right_service	not supported

CONF_HD_RECORDING_REPORT

Tag code	NumDes	Message	SNMP Support
0x0a1c	yes (cam)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get recording report from actual primary recording on a cam,
Write	p_octet	access_right_service	not supported

This comand reads the recording report of an actual running recording of a cam. The recording report contains information about RTP and VDP packets as well as VDP allocation of a running recording. The cam is specified by the num parameter.

Version 1

16 32

start_counting_time 4 Bytes
VDP_allocation_no_wait 4 Bytes
VDP_allocation_wait 4 Bytes
VDP_allocation_failed 4 Bytes

8 24

start_counting_time

time, since when the VDP allocation counting was started, this should be the time when the recording had started or since the last clear of the report.

VDP_allocation_no_wait

number of allocated VDPs since start_counting_time without blocking

VDP_allocation_wait

number of allocated VDPs since start_counting_time with blocking a time

VDP_allocation_failed

number of VDPs that couldn't allocated, because waiting for the at the block pool timed out or failed. Means also an rtp packet had to be discarded.

Version 2

16 32

Version 1 Byte	**Reserved** 3 Bytes
Start Seconds 4 Bytes	
Start Milliseconds 4 Bytes	
VDP Allocation No Wait 4 Bytes	
VDP Allocation Wait 4 Bytes	
VDP Allocation Fail 4 Bytes	
Encoder Data 48 Bytes ...	
Storage Data 48 Bytes ...	
Network Data 48 Bytes ...	
Reserved 48 Bytes ...	

8 24

Version

Version information.

Start Seconds

Timestamp in seconds since 2000 when the counting of recording data started.

Start Milliseconds

Milliseconds of the timestamp when the counting of recording data started.

VDP Allocation No Wait

Number of VDP allocation operation performed without waiting.

VDP Allocation Wait

Number of VDP allocation operation performed with waiting.

VDP Allocation Fail

Number of VDP allocation operation failed.

Encoder Data

16	32
Video RTP Packet Count 4 Bytes	
Video Byte Count 4 Bytes	
Reserved 4 Bytes	
Audio RTP Packet Count 4 Bytes	
Audio Byte Count 4 Bytes	
Reserved 4 Bytes	
Meta Packet Count 4 Bytes	
Meta Byte Count 4 Bytes	
Reserved 4 Bytes	
Reserved 12 Bytes ...	
8	24

Video RTP Packet Count

Number of video RTP packets that are deliverd to the recording.

Video Byte Count

Number of video bytes that are deliverd to the recording.

Audio RTP Packet Count

Number of audio RTP packets that are delivered to the recording.

Audio Byte Count

Number of audio bytes that are deliverd to the recording.

Meta RTP Packet Count

Number of meta RTP packets that are delivered to the recording.

Meta Byte Count

Number of meta bytes that are deliverd to the recording.

Storage Data

16	32
(Video + Audio) VDP Packet Count 4 Bytes	
(Video + Audio) Byte Count 4 Bytes	
Reserved 4 Bytes	
Audio VDP Packet Count 4 Bytes	
Audio Byte Count 4 Bytes	
Reserved 4 Bytes	
Meta VDP Packet Count 4 Bytes	
Meta Byte Count 4 Bytes	
Reserved 4 Bytes	
Reserved 12 Bytes ...	
8	24

Video VDP Packet Count

Number of video + audio VDP packets that are written to storage (since fw 4.0 there are vdp packets containing video and audio data mixed, these packets will be counted here and not in the "Audio VDP Packet Count" field).

Video Byte Count

Number of video + audio bytes that are written to storage (since fw 4.0 there are vdp packets containing video and audio data mixed, these packet bytes will be counted here and not in the "Audio Byte Count" field).

Audio VDP Packet Count

Number of audio VDP packets that are written to storage.

Audio Byte Count

Number of audio bytes that are written to storage.

Meta Packet Count

Number of meta VDP packets that are written to storage.

Meta Byte Count

Number of meta bytes that are written to storage.

Network Data

16 32

Bytes Read 4 Bytes
Bytes Write 4 Bytes
Reserved 40 Bytes ...

8 24

Bytes Read

Number of Bytes that are read from storage.

Bytes Write

Number of Bytes that are written to storage.

CONF_HD_RECORDING_REPORT_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4f	yes (cam)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get recording report from actual secondary recording on a cam, payload same as in command CONF_HD_RECORDING_REPORT
Write	p_octet	access_right_service	not supported

CONF_RECORDING_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0a9b	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	get the current recording status,	
Write	p_octet	access_right_service	not supported	

Request Structure

16		32
Camera 1 Byte	**Recording Index** 1 Byte	**Reserved** 50 Bytes ...
...		

8 24

Camera

The camera index.

Recording Index

Values:

Primary	1
Secondary	2

Response Structure

16			32
Camera 1 Byte	**Recording Index** 1 Byte	**Status** 1 Byte	**Error** 1 Byte
Current Recording Span 8 Bytes ...			
Flags 1 Byte	**Reserved** 3 Bytes		
Datarate 4 Bytes			
packet alloc no wait 4 Bytes			
packet alloc wait 4 Bytes			
packet alloc failed 4 Bytes			
recent time period 4 Bytes			
packets recorded high 4 Bytes			
packets recorded low 4 Bytes			
packets lost high 4 Bytes			
packets lost low 4 Bytes			

8 24

Camera

The camera index.

Recording Index

Values:

Primary	1
Secondary	2

Status

Values:

OFFLINE	1	configured for not recording
IDLE	2	recorder is up and running on internal ram but no storage is configured for recording
PEND	3	trying to connect storage or waiting for block list from vrm
RUNNING	4	storage connected and recording running or prepared for recording
ERROR	5	error on storage ocurred
REC_SRC_ERROR	6	error of recorder source(encoder)

Error

See error codes of CONF_SPAN_USE_STATUS.

Current Recording Span

The address of the span the camera is recording to.

16		32
Target ID 4 Bytes		
Target Index 1 Byte	**Lun** 1 Byte	**Span Index** 2 Byte
8	24	

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target Index

In index of the iscsi target.

Lun

The lun indentifier.

Span Index

The index of the span.

Flags

Values:

ENCRYPTION	0x01	encrypted recording

Datarate

The datarate that is written to the storage.

packet alloc no wait

recently alloc no wait vdp packets (1.5 kbyte) since recent time period

packet alloc wait

recently alloc wait vdp packets (1.5 kbyte) since recent time period (indicates problems of the recording get its data to the drive)

packet alloc failed

recently alloc failed vdp packets (1.5 kbyte) since recent time period (indicates problems of the recording get its data to the drive, corrupt data stream)

recent time period

time period in which the packet allock counter were counted

packets recorded high

high part (32 bits network order) of a 64 bit counter for all recorded packets of this recording since device start up (packets counted by "packet alloc no wait" and "packet alloc wait" not included)

packets recorded low

low part (32 bits network order) of a 64 bit counter for all recorded packets of this recording since device start up (packets counted by "packet alloc no wait" and "packet alloc wait" not included)

packets lost high

high part (32 bits network order) of a 64 bit counter for all packets, which couldn't be recorded by this recording since device start up (packets counted by "packet alloc failed" not included)

packets lost low

low part (32 bits network order) of a 64 bit counter for all packets, which couldn't be recorded by this recording since device start up (packets counted by "packet alloc failed" not included)

CONF_RECORDING_RETENTION_TIME

	Tag code	NumDes	Message	SNMP Support
	0x0a30	yes (camera)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the primary recording retention time for a camera (in seconds).
Write	t_dword	access_right_service	set the primary recording retention time for a camera (in seconds) (value >= 1009152000) means maximum, no influence on local recording, only for vrm managed recording. (effect takes place on next mounted span)

CONF_RECORDING_RETENTION_TIME_SECONDARY

	Tag code	NumDes	Message	SNMP Support
	0x0a48	yes (camera)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the secondary recording retention time for a camera (in seconds).
Write	t_dword	access_right_service	Description : set the secondary recording retention time for a camera (in seconds) (value >= 1009152000) means maximum, no influence on local recording, only for vrm managed recording. (effect takes place on next mounted span)

CONF_MAX_RECORDING_RETENTION_TIME

Tag code	NumDes	Message	SNMP Support
0x0b5b	yes (camera)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the max recording retention time for a camera (in seconds), see detailed description
Write	p_octet	access_right_service	set the max recording retention time for a camera (in seconds), see detailed description

Payload Structure

16	32
max retention time 4 Bytes	
rec idx 2 Byte	**reserved** 2 Byte
8	24

max retention time

max retention in seconds, 0 or values bigger than about 32 years (value >= 1009152000) means maximum, in case of in payload for read direction, this field can be ignored and treaded as reserved field

rec idx

recording index

Values:

primary recording	1
secondary recording (active low)	2

CONF_RECORDING_MAX_RETENTION_TIME

This command sets the max retention time for recording identified by the rec idx and the camera in the num parameter starting from 1. For read direction a inpayload is required with same format as described above, except the max retention time field can be ignored. The max retention time will be stored as absolute local time in seconds since 2000 in each span unit header. It will be used to clear a recording span when this time expires. If a max retention time is configured unequal 0 and less than maximum, the recording time on a span is limited to a one day time span. If this limit is reached by the recording, a span switch will be triggered.

CONF_REMOTE_REC_DEVICE

Tag code	NumDes	Message	SNMP Support
0x0a85	yes (max 8)	no	no

	Datatype	Access Level	Description
Read	p_string	access_right_minimal	Get the ip address or iqn of the remote recording device (max. 50 characters).
Write	p_string	access_right_service	Set the ip(ipv4 only) address or the iqn (starting with "iqn." or * for wildcard) of a remote recording device (max. 50 characters) on one entry ore use any free entry by num = 0, the ip/iqn will be not added in that case if it is already in the list. To clear a entry send a empty string(zero termination only) on a valid index(1 - 8)

CONF_REC_MGNT

Tag code	NumDes	Message	SNMP Support
0x0a89	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	Read the type of the recording management (0 = LOCAL, 1 = VRM, 2 = VRM+ANR, 3 = ONLY LOCAL (obsolete), 4 = DUAL VRM, 5 = ONVIF)
Write	t_octet	access_right_service	Set the type of the recording management (0 = LOCAL, 1 = VRM, 2 = VRM+ANR, 3 = ONLY LOCAL(obsolete), 4 = DUAL VRM, 5 = ONVIF) (not possible to change settings while recording)

CONF_HD_MGR_REC_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0aae	yes (cam)	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	see detailed description	
Write	p_octet	access_right_service	not supported	

Payload Structure

8	16	24	32
rec state 1 Byte	**rec preset** 1 Byte	**enc preset** 1 Byte	**flags** 1 Byte

rec state

state of the recording, state OFF not in payload of msg

Values:

OFF	0
NO RECORDING	1
STAND BY	2
PRE ALARM RECORDING	3
ALARM RECORDING	4
POST ALARM RECORDING	5

rec preset

actual used recording preset from 1 to 10 or 0 if no preset is used

enc preset

actual used encoder preset from 1 to 8 or 0 if no preset is used

flags

these flags show the alarm states and the recording mode

Values:

alarm recording mode	0x01
input alarm	0x02
motion alarm	0x04
video loss	0x08
virtual alarm	0x10
reserved for extension	0x80

CONF_HD_MGR_REC_STATUS

This command shows the state for a recording. It can be read and it will be send at state changes. State changes are the change of rec state or rec preset only. Within the Msg there is no distiction between the state OFF and NO RECORDING. That means no msg will be send on the state change between this two states. The msg will never contain the state OFF but the read response maybe. OFF state means the recording is configured to off by set it to stop. NO RECORDING can be caused by many things e.g. no recording on the schedule, no storage present and so on. STAND BY means there is recording on the schedule but not at the moment. The recording scheduler waits for the time to start the recording. All the other states indicating a running recording. In case of alarm recording in prealarm state, the message state will be PRE ALARM RECORDING even if the pre alarm recording takes place in the ram only, if at least the storage is connected. If there is no storahge connected and recording is configured for running, the state will alway be NO RECORDING.

CONF_HD_MGR_REC_STATUS_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0aaf	yes (cam)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	same as CONF_HD_MGR_REC_STATUS but for secondary recording (see cmd CONF_HD_MGR_REC_STATUS)
Write	p_octet	access_right_service	not supported

CONF_HD_FILE_INFO

Tag code	NumDes	Message	SNMP Support
0x091d	no	yes	no

	Datatype	Access Level	Description
Read	void	access_right_user	read not supported (only for messages),
Write	void	access_right_service	not supported

Payload Structure

16 32

start time 32 Bits		
end time 32 Bits		
flags 32 Bits		
file id 32 Bits		
target id 4 Bytes		
target idx 1 Byte	**lun** 1 Byte	**span idx** 16 Bits
cam 1 Byte	**recording idx** 1 Byte	**reserved** 2 Bytes

8 24

start time

start time in seconds since 2000 of the recording file

end time

end time in seconds since 2000 of the recording file

flags

file info flags, see CONF_HD_PARTITION_FILE_INFO

file id

File ID on the file unique per span

target id

target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

target idx

target index of the span

lun

lun of the span

span idx

span index

cam

camera from 1 to ...

recording idx

recording index (primary or secondary)

Values:

primary	1
secondary	2

CONF_HD_FILE_INFO

This message is used for updating actual recording files, it will be send periodically about every second by the recording device, that is running the recording on that file. The message informs about the start and end time changes of an file. For uniqe idetification, the message includes the span address and a file id. it also has the information about the recording cam (attention: in older firmware the file info msg only has the first 4 payload fields(from start time to file id))

CONF_HD_FILE_INFO_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a64	no	yes	no

	Datatype	Access Level	Description
Read	void	access_right_user	read not supported (only for messages), see definition CONF_HD_FILE_INFO
Write	void	access_right_service	not supported

CONF_HD_RELOAD_PARTITION_FILE_INFO

Tag code	NumDes	Message	SNMP Support
0x091e	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported (message signals an obsolete file list (num==partition))
Write	void	access_right_service	not supported

CONF_HD_RECORDING_ACTIVE

Tag code	NumDes	Message	SNMP Support
0x0908	yes cam	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	1: yes 0: no (message only, a request on this command always returns zero)
Write	-	access_right_service	not supported

CONF_HD_REPLAY_ENCRYPTED_DATA

	Tag code	NumDes	Message	SNMP Support
	0x0c89	no	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	not supported (message signals occurence of encrypted data on a replay session that cannot be decrypted, if the state keeps on, the message however is limited to max. 1 per second)	
Write	void	access_right_service	not supported	

CONF_HD_REPLAY_AUTHENTICITY_OUT

	Tag code	NumDes	Message	SNMP Support
	0x0c9f	Message : yes	Description : not supported (message reports the authenicity check results of an running replay session) ##desc:./doc/CONF_HD_REPLAY_AUTHENTICITY_OUT.htm##	no
	Datatype	**Access Level**	**Description**	
Read	%	access_right_minimal	%	
Write	void	access_right_user	not supported	

CONF_START_RECORD

	Tag code	NumDes	Message	SNMP Support
	0x0acc	yes (1 - primary recording, 2 - secondary recording)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	Start recording on choosen cams, payload needs a variable count pairs of 32 bit fiels for cam state (posibble states: 1 : on, 0 : off) and 32 bit fiels for cam selection mask, this mask will layed over the state bit field and can be used to choose the cams specific cams, other cams will be ignored, each bit field pair represents 32 cams: first pair - cam 1 to cam 32, second pair -cam 33 to cam 64 and so on	
Write	p_octet	access_right_service	Start recording on choosen cams, payload needs a variable count pairs of 32 bit fields for cam state (posibble states: 1 : on, 0 : off) and 32 bit fiels for cam selection mask, this mask will layed over the state bit field and can be used to choose the cams specific cams, other cams will be ignored, each bit field pair represents 32 cams: first pair - cam 1 to cam 32, second pair -cam 33 to cam 64 and so on	

CONF_SET_REC_BUFFER_SIZE

Tag code	NumDes	Message	SNMP Support
0x0ae1	yes (line)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get size of recording buffer in bytes (Bosch streaming gateway only) first 4 bytes : DWORD rec_idx second 4 bytes : DWORD size in bytes(return direction)
Write	p_octet	access_right_service	set size of recording buffer in bytes (restart of recording nessessary) (Bosch streaming gateway only, default 5 MB) first 4 bytes : DWORD rec_idx second 4 bytes : DWORD size in bytes

CONF_AUDIO_REC_FORMAT

Tag code	NumDes	Message	SNMP Support
0x0ae9	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	get audio format for recording (0: NO, 1: G711, 2: L16, 3: AAC)
Write	t_octet	access_right_service	set audio format for recording (0: NO, 1: G711, 2: L16, 3: AAC) (running recording has to be stopped and restarted in order to become effective)

CONF_MANAGING_VRM

Tag code	NumDes	Message	SNMP Support
0x0aeb	yes (1 - primary recording, 2 - secondary recording)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_service	see detailed description

This command can be used to set or read the IP address, port and user and to set the password of the managing VRM and the backup VRM. The command cannot be used for reading the VRM password, it will return the string "********" instead.

Payload Structure

16 32

ip 4 Bytes		
port 16 Bits	**flags** 8 Bits	**reserved** 1 Byte
user 32 Bytes ...		
password 32 Bytes ...		
backup ip 4 Bytes		
backup port 16 Bits	**reserved** 16 Bits	

8 24

ip

IP address of the managing VRM

port

Port of the managing VRM

flags

Additional flags

Values:

USE_SSL 0x01

user

VRM user: Max 31 ASCII character string with zero termination. If the user is shorter than 31 characters, the remaining bytes need to be filled up with 0x00 values.

password

VRM password: Max 31 ASCII character string with zero termination. If the password is shorter than 31 characters, the remaining bytes need to be filled up with 0x00 values.

backup ip

IP address of the backup VRM

backup port

Port of the backup VRM

CONF_REC_STORAGE_REQ_CFG

Tag code	NumDes	Message	SNMP Support
0x0b30	yes (recording index: 1 - primary, 2 - secondary)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read error tollerance parameter for storage requests from recording (4 bytes, 2 words) (1st word retry in network order 0xffff for retry always, 2nd word timeout seconds in network order 0 for default timeout 0xffff for never timeout
Write	p_octet	access_right_service	set error tollerance parameter for storage requests from recording (4 bytes, 2 words) (1st word retry in network order 0xffff for retry always, 2nd word timeout seconds in network order 0 for default timeout 0xffff for never timeout

CONF_RECORDING_BUFFER_LEVEL

Tag code	NumDes	Message	SNMP Support
0x0b70	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	returns the levels in percent of the recording rate control based on the recording buffer fill-level: 4 bytes: level off (if the fill level is below that value the rate control will be turned off), 4 bytes: level on (if the fill level is above that value the rate control will be turned on), (set both values to 0 to disable the rate control). This features is enabled per default on TI devices with 10% (off) and 30% (on)
Write	p_octet	access_right_service	returns the levels in percent of the recording rate control based on the recording buffer fill-level: 4 bytes: level off (if the fill level is below that value the rate control will be turned off), 4 bytes: level on (if the fill level is above that value the rate control will be turned on), (set both values to 0 to disable the rate control). This features is enabled per default on TI devices with 10% (off) and 30% (on)

CONF_HD_REC_BUFFER

Tag code	NumDes	Message	SNMP Support
0x0b8f	no	no	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	void	access_right_service	not supported

In Payload Structure

16	32
cam 2 Byte	**rec idx** 2 Byte
8	24

Reply Payload Structure

16	32
cam 2 Byte	**rec idx** 2 Byte
full buffer bytes 4 Bytes	
allocated buffer bytes 4 Bytes	
min buffer bytes 4 Bytes	
filled buffer bytes 4 Bytes	
incommming data rate kBits per sec 4 Bytes	
buffer time from utc 4 Bytes	
buffer time to utc 4 Bytes	
buffer time recording utc 4 Bytes	
recent buffer replay time utc 4 Bytes	
buffer replay byte cnt high 4 Bytes	
buffer replay byte cnt low 4 Bytes	
buffer replay cnt since utc 4 Bytes	
buffer replay cnt since ms offset 2 Byte	**reserved** 2 Byte
buffer replay data rate kBits per sec 4 Bytes	
8	24

cam

camera starting with 1 (in payload)

rec idx

recording index (in payload)

Values:

primary recording	1
secondary recording (active low)	2

full buffer bytes

full amount of buffer in bytes for recording on this device

allocated buffer bytes

actual allocated buffer for this recording

min buffer bytes

the minimum amount of buffer the recording would allocate

filled buffer bytes

the amount of buffer that contains data for recording (not the data which are recorded but still in buffer)

incommming data rate kBits per sec

incomming datarate in record buffer in KBit per sec (avarange over last seconds)

buffer time from utc

time stamp of oldest record data in buffer in seconds since 2000 utc

buffer time to utc

time stamp of latest record data in buffer in seconds since 2000 utc

buffer time recording utc

time stamp of record data in buffer to be recorded next in seconds since 2000 utc

recent buffer replay time utc

timestamp of last data requested by a buffer replay in seconds since 2000 utc

buffer replay byte cnt high

64 bit byte counter of replayed buffer replay data (upper 32 bit)

buffer replay byte cnt low

64 bit byte counter of replayed buffer replay data (lower 32 bit)

buffer replay cnt since utc

time in seconds since 2000 utc, since when the replay byte cnt counter is counting

buffer replay cnt since ms offset

milli seconds offset of the byte cnt counter time

buffer replay data rate kBits per sec

buffer replay datarate from record buffer in KBit per sec (avarange over last seconds)

CONF_HD_REC_BUFFER

This command gets information about the recording buffer size, it includes the full amount of buffer for all recordings, the allocated, and the minimum amount of buffer of the recording specified by the in payload. The "full buffer bytes" is usually preallocated memory at startup and won't change it's size as long as the device is running and is the same for all recordings (except streaming gateway). The "min buffer bytes" is the amount of buffer the device will at least allocate if running. The "allocated buffer bytes" is the amount of buffer the recording has actually allocated. This value can be higher than "min buffer bytes", if nt all possible recrdings are running. The unused buffer of non running recordings will be used by the running ones. If the recording is deaktivated, the value will be 0. When running, the value wil be usually at least "min buffer bytes", but it can be temporary less than that value, if e.g. the recording was started shortly before and another already running recording has allocated the whole buffer. In that case it could take some time free the buffer by the running recording, so the allocated buffer will increase over the time. The "filled buffer bytes" is the amount of buffer memory of the the "allocated buffer bytes", which contains data for recording or in case of alarm recording, the amout to be recorded, if an alarm would occur in this moment. The "filled_buffer bytes" can therefor be used to calculate the fill level of the recording buffer. The information also includes the incomming data rate, which is the data rate of data from the video, audio, ... encoders to the record buffer. Also th time range of available data in buffer are presented along with the progress of the recording, all in seconds since 2000 utc. This is usefull to observe backlog of recording. Internally there are replay sessions running directly on the record buffer, therefor information about the buffer replay are also present, including the utc timestamp of the last replay data from buffer and byte counter and replay data rate.

CONF_MAX_GOP_LENGTH_VALUE

Tag code	NumDes	Message	SNMP Support
0x0b9d	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	Read the max gop length for recording
Write	t_dword	access_right_service	Set the max gop length for recording, 0 means back to default

CONF_ALARM_BACKUP_REC_SPEED_LIMIT

Tag code	NumDes	Message	SNMP Support
0x0bb0	line (1 ...n)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_user	get the speed limit parameter, see detailed description
Write	p_octet	access_right_service	set the speed limit parameter, see detailed description

Payload Structure

16	32
speed limit 4 Bytes	
data rate limit 4 Bytes	
8	24

speed limit

speed limit in percent (0 to 9999, default 200)

data rate limit

data rate limit in Kbps (0 - no limit, default 0)

CONF_ALARM_BACKUP_REC_SPEED_LIMIT

This command can be used to limit the data traffic caused by backuped recording per line (num parameter 1 to n). This can be done by limiting the data rate and/or the speed in percent(100 percent means backup speed like replay in normal speed), based on the time information of the video data. The backup can be stopped completly, by sending a speed limit by 0 and continued again by sending a non zero speed later. It cannot be stopped by the data rate, a value of 0 for the data rate means to disable the limit for the data rate. The data rate parameter isn't applied immediately, but when the next backup starts.

CONF_BACKUP_RECORDING_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0bb4	line (1 ...n)	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_user	get the backup recording status, see detailed description	
Write	p_octet	access_right_service	write not supported, see detailed description	

Payload Structure

16	32
rec idx life 2 Byte	**rec idx backup** 2 Byte
free MB 4 Bytes	
backuped MB 4 Bytes	
unbackuped MB 4 Bytes	
lost counter 4 Bytes	
msg trigger 1 Byte	**reserved** 3 Byte
8	24

rec idx life

Recording index of the recording, which is configured for life data recording (0 - none,1 - primary, 2 - secondary)

rec idx backup

Recording index of the recording, which is configured for data backup of the life data recording (0 - none,1 - primary, 2 - secondary)

free MB

Free Storage space in MB for the life data recording

backuped MB

Storage space in MB with records of the life data recording for this line, which are already backuped or which are not required to be backuped

unbackuped MB

Storage space in MB with records of the life data recording for this line, which have to be backuped or potentially required to be backuped, because they are in range of the max pre alarm time

lost counter

Counter for events of records, which had to be backuped and that couldn't be backuped, on deleting or overwriting by newer records

msg trigger

reason for sending this status

Values:

REQUESTED	0	response on request
ALL_CLEAR	1	all clear threshold passed
WARNING	2	warning threshold passed
DATA_LOSS	3	new data loss

CONF_BACKUP_RECORDING_STATUS

This command returns the backup recording status on a line, specified by the num parameter (1 ... n). First it looks for a valid configuration of backup recording. Valid means, there exists a running recording configured to record the life data and a second recording for backup, which copies the data from the life data recording storage to it's recording storage on alarm or continuously. If both "rec idx life" and "rec idx backup" are none zero, there is a valid configuration. Then the following fields inform about the status and progress of the backuped recording. The utilization of the life data recording storage can be calculated in percent with the equation: utilization = ("unbackuped MB" * 100)/("unbackuped MB" + "backuped MB" + "free MB"). The status will be send as message, if the utilization will cross some thresholds. There are two thresholds, the "All Clear" threshold and the "Warning" threshold. These thresholds can be configured with the command CONF_BACKUP_RECORDING_STATUS_MSG_THRESHOLD (see rcp documentation). Another reason for sending this message will be on deleting or overwriting of records, which had to be backuped but weren't backuped before deletion, or were deleted while still within the maximum retention time range. Every time this event happens, the lost counter will be incremented. This counter will count since device boot.

CONF_BACKUP_RECORDING_STATUS_MSG_THRESHOLD

	Tag code	NumDes	Message	SNMP Support
	0x0bb5	line (1 ...n)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_user	get the thresholds, see detailed description	
Write	p_octet	access_right_service	set the thresholds, see detailed description	

Payload Structure

16		32
all clear threshold 1 Byte	**warning threshold** 1 Byte	**reserved** 2 Byte
8	24	

all clear threshold

threshold for all clear in percent

warning threshold

threshold for warning in percent

CONF_BACKUP_RECORDING_STATUS_MSG_THRESHOLD

This command configures the thresholds for the CONF_BACKUP_RECORDING_STATUS message. The message will be send, when the storage space utilization will cross the "all clear" threshold by changed utilization from higher to lower utilization or when crossing the "warning" threshold by changed utilization from lower to higher utilization.

CONF_BUFFERED_RECORDING_MODE

Tag code	NumDes	Message	SNMP Support
0x0bcd	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	get the buffered recording mode: 0 off, 1 on
Write	t_octet	access_right_service	set the buffered recording mode: 0 off, 1 on,

Payload Structure

Buffered Rec Mode
1 OCTET

8

Buffered Rec Mode

manual configuration	0
buffered recording configuration mode	1

CONF_BUFFERED_RECORDING_MODE

This command can be used to configure the buffered recording configuration mode. The default is the manual mode, which is used for standard (single/dual) recording or manually configured buffered recording, which means the primary and secondary recording will be setup to do buffered recording. In case of buffered recording configuration mode, the device recording configuration will be changed internally to an buffered recording setup. The recording will then be configured like a single primary recording via recording profiles but internally the configuration is mapped to a coresponding buffered recording setup. When set to this mode, the secondary recording isn't accessable anymore via recording profile configuration (affected comands: CONF_HD_RECORD_PROFILE, CONF_HD_RECORD_PROFILE_V2, CONF_HD_RECORD_PROFILE_SECONDARY, CONF_HD_PROFILE_V2_SECONDARY) and start/stop configuration(affected commands: CONF_START_RECORD, CONF_START_SPAN_RECORD, CONF_HD_MGR_START, CONF_HD_MGR_START_SECONDARY, CONF_HD_MGR_STOP, CONF_HD_MGR_STOP_SECONDARY). When the mode is changed, the actual recording is stopped.

CONF_WPS_BUTTON_ENABLED

Tag code	NumDes	Message	SNMP Support
0x0be8	no	no	no

	Datatype	Access Level	Description
Read	flag	access_right_user	Returns if the WPS button is enabled (1) or not (0). Default is enabled.
Write	flag	access_right_service	Enables or disabled the WPS button. When the button is diabled, pressing the button has no effect.

CONF_HD_FORMAT_STORAGE_ERROR_LOG_CNT

Tag code	NumDes	Message	SNMP Support
0x0c65	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	returns the number of failed local storage formats
Write	t_dword	access_right_service	write not supported

CONF_SD_CARD_LIFE_SPAN_STATUS

Tag code	NumDes	Message	SNMP Support
0x0c5A	SD card slot ID. Starts with 1.	yes	yes

	Datatype	Access Level	Description
Read	p_octet	access_right_user	see detailed description
Write	void	access_right_service	not supported

Payload Structure

16 32

length 2 Byte	Tag 2 Byte
Payload 4 Bytes	

8 24

length

Length of the tag incuding the 4 header bytes

Values:

Tag

ID of parameter

Values:

Payload

Value of parameter

Values:

Tagged format

The format has an internal tagged structure with the following tags:

MANF_ID	1
PROD_ID	2
PROD_STRING	3
NUM_BLOCKS	4
ESTIMATED_LIFE_SPAN_PERCENT_DONE	5
MANF_NAME	6
SERIAL_NO	7

MANF_ID

SD Card manufacturer ID.

length = 5 2 Byte	tag = 1 2 Byte
Manufacturer ID 1 Bytes	

16 32

PROD_ID

SD Card product ID.

length = 6 2 Byte	tag = 2 2 Byte
Product ID 2 Bytes	

16 32

PROD_STRING

SD Card product string.

length 2 Byte	tag = 3 2 Byte
Product string Length - 4 Bytes	

16 32

NUM_BLOCKS

Number of SD Card 512 byte blocks.

length = 8 2 Byte	**tag = 4** 2 Byte
Number of blocks 4 Bytes	

16 32

ESTIMATED_LIFE_SPAN_PERCENT_DONE

Consumed value of estimated lifespan for SD-Card in percent (only for supported cards)

length = 5 2 Byte	**tag = 5** 2 Byte
percent value 1 Bytes	

16 32

MANF_NAME

Manufacturer plain text name (only for supported cards)

length 2 Byte	**tag = 6** 2 Byte
Manufacturer name string Length - 4 bytes	

16 32

SERIAL_NO

SD-Card serial number

length = 8 2 Byte	**tag = 7** 2 Byte
Serial number 4 Bytes	

16 32

CONF_SD_CARD_LIFE_SPAN_ALARM_THRESHOLD

Tag code	NumDes	Message	SNMP Support
0x0c87	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	Read the current alarm threshold when SD-Cards should trigger an EndOfLifespan alarm: 0..100%; 0 = off (Only for SD-Card which supports this feature)
Write	t_octet	access_right_service	Write the current alarm threshold when SD-Cards should trigger an EndOfLifespan alarm: 0..100%; 0 = off (Only for SD-Card which supports this feature)

CONF_SD_CARD_LIFE_SPAN_MANF

Tag code	NumDes	Message	SNMP Support
0x0cde	yes	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_user	Read SD-Card manufacturer for SD Lifespan indicator. 0=Autodetect; 1=SANDISK; 2=SONY; 3=MICRON; 4=EMMC_GENERIC
Write	t_octet	access_right_service	Write SD-Card manufacturer for SD Lifespan indicator. 0=Autodetect; 1=SANDISK; 2=SONY; 3=MICRON; 4=EMMC_GENERIC

CONF_SD_CARD_LIFE_SPAN_ENABLE

Tag code	NumDes	Message	SNMP Support
0x0c9c	no	no	no

	Datatype	Access Level	Description
Read	f_flag	access_right_user	Read enable state for SD-Card live-span information (may be disabled for SD-Card which does not supports this feature)
Write	f_flag	access_right_service	Write enable state for SD-Card live-span information (may be disabled for SD-Card which does not supports this feature)

CONF_NBR_OF_SD_CARD_SLOTS

Tag code	NumDes	Message	SNMP Support
0x0cb2	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get number of sd card slots
Write	%	access_right_minimal	%

CONF_EXT_RECORDER_BITRATE_STATISTICS

Tag code	NumDes	Message	SNMP Support
0x0c94	-	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	recording bitrates:
Write	p_octet	access_right_service	not supported

In Payload Structure

16 32

cam 2 Byte		**rec idx** 2 Byte
data type 1 Byte	**re -** .	**served** 3 Byte

8 24

cam

camera starting with 1

rec idx

recording index

Values:

primary recording	1
secondary recording	2

data type

Values:

video data	0
other data	1

Reply Payload Structure

See reply payload of CONF_EXT_ENCODER_BITRATE_STATISTICS

CONF_HD_REPLAY_START

Tag code	NumDes	Message	SNMP Support
0x0902	no	yes	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	returns t_int parameter in percent of realtime replay (default +100%); 0 if suspended or stopped; Session ID is needed
Write	t_int	access_right_user	start a HD replay at the current position; t_int parameter in percent of realtime replay (default +100%); negative values will result in a reverse replay; Session ID is needed

CONF_HD_REPLAY_START_EX

Tag code	NumDes	Message	SNMP Support
0x0c74	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported, use CONF_HD_REPLAY_START
Write	p_octet	access_right_user	start a HD replay at the current position; 32 bytes: 4Bytes int parameter in percent of realtime replay (default +100%); negative values will result in a reverse replay; 1byte flags, 1 byte cseq, remaining reserved, Session ID is needed

CONF_HD_REPLAY_STOP

Tag code	NumDes	Message	SNMP Support
0x0903	no	yes	no

	Datatype	Access Level	Description
Read	flag	access_right_minimal	1=replay is stopped; 0=replay is in progress
Write	flag	access_right_user	stop a current HD replay; replay pointer will not be affected; Session ID is needed

CONF_HD_REPLAY_STOP_TIME

Tag code	NumDes	Message	SNMP Support
0x0904	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_user	see detailed description

Payload Structure

16	32
Seconds 4 Bytes	
Milliseconds 4 Bytes	
8	24

Seconds

Absolute time in seconds since 1.1.2000 00:00h.

This command will return a write error if the timeposition is outside a recording set. The Session ID is needed. A value of zero clears the stop marker.

CONF_HD_REPLAY_SEEK_TIME

Tag code	NumDes	Message	SNMP Support
0x0905	no	yes (every second)	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_user	see detailed description

Payload Structure

16	32
Seconds 4 Bytes	
Rtp Timestamp 4 Bytes	

8 24

Seconds

Absolute time in seconds since 1.1.2000 00:00h.

Rtp Timestamp

Only in message. Rtp timestamp of the first replayed RTP packet of this second.

extended payload Structure (optional, instead of above payload structure; for write direction only)

16		32
seconds 4 Bytes		
miliseconds 2 Byte	**flags** 1 Byte	**payload** 1 Byte
conf_roi: hPos 2 Byte	**conf_roi: vPos** 2 Byte	
conf_roi: hSize 2 Byte	**conf_roi: vSize** 2 Byte	
reserved 4 Byte		
reserved 4 Byte		
reserved 4 Byte		
reserved 4 Byte		

8 24

seconds

Absolute time in seconds since 1.1.2000 00:00h.

miliseconds

miliseconds

flags

Values:

Bit 0	Time Zone	payload contains the timezone as quarter hours offset (signed char)
Bit 1	Iframe Preview	sends the Iframe on the seek position (transcoder case only)
Bit 2	Sec Accuracy	seek time accuracy to seconds (transcoder case only)

payload

one payload byte, conntend depends on the flags field

conf_roi

select region of interest hPos,vPos,hSize,vSize (each entry 2 bytes): starting left upper edge, each 2bytes 0..32768, vSize==0 means keep aspect ratio

This command will return a write error if the timeposition is outside a recording set. The Session ID is needed.

CONF_HD_REPLAY_SEEK_IFRAME

Tag code	NumDes	Message	SNMP Support
0x0907	no	no	no

	Datatype	Access Level	Description
Read	t_int	access_right_minimal	obsolete
Write	t_int	access_right_user	set the replay pointer to the beginning of the next frame/last I-Frame(s); parameter t_int -1 back to last iframe, 1 to next frame; Session ID is needed

CONF_HD_REPLAY_EVENT_INFO

Tag code	NumDes	Message	SNMP Support
0x091f	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_user	read only

This command returns a list of recorded alarm events in backward order, since fw 4.0 events are only alarm state changes. Max size of the list is 128 entries. The command needs a valid replay session. The events start from the given end time (first 4 bytes of payload) in secconds since 2000. If the payload consists of at least 8 bytes and the bytes 5-8 are filled with the value 0xeeeeeeee the extended event structure is used in the response payload. Payload size with all optional parameters is 12 bytes (see "In Payload"). The response contains a flag field, containing the alarm flags, which give information about the alarm, which are activated at this event or if all alarms went to off state, if no one of the alarm flags were set. the flags also contains the time zone information with sign and number of quarter hours.

In Payload

16 32

search begin time 4 Bytes
special field 4 Bytes
earliest time 4 Bytes

8 24

Payload Structure

16 32

n Events n*m Bytes

8 24

Event Structure

Simple Event Structure

16 32

Event Time 4 Bytes
Flags 4 Bytes

8 24

Extended Event Structure

16 32

Upper X 1 Byte	**Upper Y** 1 Byte	**Lower X** 1 Byte	**Lower Y** 1 Byte
Event Time in secs 4 Bytes			
residual milliseconds 2 Bytes		**reserved** 2 Bytes	
Flags 4 Bytes			

8 24

search begin time

Start time of the search in seconds since 2000. The search will procceed into the past.

special field

Optional parameter, if set to 0xeeeeeeee, the reply will contain data in extended event structure format. Any other value has no meaning yet.

earliest time

Optional parameter, Erliest time in seconds since 2000. The search won't search in record files, which lie before that time. The search will be performed on file which lie completely or partialy within the search interval of earliest time and search begin time. If not set, the default will be 0.

Event Time

Seconds since 2000.

Flags

Values:

Bit 0	Motion Alarm

Bit 1	Input Alarm
Bit 2	Video Loss
Bit 3	Virtual alarm (see CONF_VIRTUAL_ALARM_STATE)
Bit 24 - 30	time zone quarter hours (mask: 0x7f000000)
Bit 31	time zone quarter hours sign (1 - negativ)

Upper X,Y and Lower X,Y

Specifies a bounding box (upper left corner and lower right corner). If an object can be associated with the alarm then the bounding box will be around this object otherwise the whole image is the bounding box. The values are normalized to the image size and are between 0 and 255. The origin of the coordinate system is the upper left corner of the image.

CONF_HD_REPLAY_PARTITION_EVENT_INFO

Tag code	NumDes	Message	SNMP Support
0x091c	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the last 512 alarm events of the partition (obsolete)
Write	p_octet	access_right_user	read only

CONF_HD_REPLAY_MOTION_SAMPLES

Tag code	NumDes	Message	SNMP Support
0x095d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_user	read only

Payload Structure

16 32

Sec. End 4 Bytes
Sample Interval 4 Bytes
Maximum Sample Bytes 4 Bytes
Bytes Per Sample 4 Bytes

8 24

Sample Interval

Values:

Seconds	0
Minutes	1
Houres	2

Maximum Sample Bytes

Values:

Seconds	0

Bytes Per Sample

1, 2, 4, 8. Time is included.

Read motion samples of a recording. Session ID required.

CONF_HD_REPLAY_FAST_INTRA_DELAY

Tag code	NumDes	Message	SNMP Support
0x095e	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the delay (in 10ms) between the frames in intra only replay mode
Write	t_dword	access_right_user	set the delay (in 10ms) between the frames in intra only replay mode

CONF_HD_REPLAY_FAST_INTRA_FPS

Tag code	NumDes	Message	SNMP Support
0x0ac2	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get max frames per second for intra only replay mode
Write	t_dword	access_right_user	set max frames per second for intra only replay mode (0 = default)

CONF_HD_REPLAY_LIVE

Tag code	NumDes	Message	SNMP Support
0x0963	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get replay connection live mode
Write	t_dword	access_right_user	set a replay connection to live mode (only supported with transcoder) 1:=on, 0:= off replay commands like start or seek terminate the live mode

CONF_HD_REPLAY_SIZE_INFO

Tag code	NumDes	Message	SNMP Support
0x0906	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	see detailed description
Write	p_octet	access_right_user	-

Request Packet

16	32
Starttime 4 Bytes	
Stoptime 4 Bytes	

8 24

Starttime

In seconds since 2000.

Stoptime

In seconds since 2000.

Reply Packet

16	32
Max Size 8 Bytes ...	
Min. Number Of Rtp Packets (will tend to be more) 4 Bytes ...	

8 24

Max Size

In Bytes.

Replay session id required. Start time and stop time has to be within one file.

CONF_HD_REPLAY_VCD_LAYER

Tag code	NumDes	Message	SNMP Support
0x09c8	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	requested vcd layer 0=no vcd data, 1=layer1 ,2=layer2,..; replay session id required
Write	t_dword	access_right_user	requested vcd layer 0=no vcd data, 1=layer1 ,2=layer2,..; replay session id required

CONF_HD_REPLAY_VCD_CONFIG_ID

Tag code	NumDes	Message	SNMP Support
0x0a5f	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	requested vcd config ID 0=all vcd data, 1=config1 data, 2=config2 data,..; replay session id required
Write	t_dword	access_right_user	requested vcd config 0=all data, 1=config1 data ,2=config2 data,..; replay session id required

CONF_HD_REPLAY_FORENSIC_SEARCH_SETUP

Tag code	NumDes	Message	SNMP Support
0x0b0a	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	setup a forensic search
Write	p_octet	access_right_user	setup a forensic search, see detailed description

This command is related to a replay session. The replay session will start a forensic search with the provided ruleengine script. The return payload consists of a unique 4 byte ForensicSearch ID which can be refound in the corresponding message (HD_REPLAY_FORENSIC_SEARCH_RESULT). In the case of an error the RCP_ERROR_COMMAND_SPECIFIC error is returned with an error code (see HD_REPLAY_FORENSIC_SEARCH_RESULT) followed by an additional text.

Error Response Structure

RCP_ERROR_COMMAND_SPECIFIC	Error type	Error text
1 Byte	1 Byte	n Byte

Error type

Name	Value
e_ForensicSearchErrorNone	0x00
e_ForensicSearchErrorScriptTooLong	0x01
e_ForensicSearchErrorInternal	0x02
e_ForensicSearchErrorNoRuleSelected	0x03
e_ForensicSearchErrorREConfiguration	0x04
e_ForensicSearchErrorSyntax	0x05
e_ForensicSearchErrorMemory	0x06

Error text

The error text is a detailed description of the error. E.g. for syntax errors the line of the script and error type are mentioned.

Payload Structure

16 32

Begin [s] 4 Byte		
End [s] 4 Byte		
reserved 4 Byte		
Script length 2 Byte	**Script mode** 1 Byte	**VCA Profile ID** 1 Byte
Layer Mode 1 Byte	**Flags** 1 Byte	**Layer mask** 2 Byte
Ruleengine script script length		
...		

8 24

Begin[s]

Defines begin of forensic search in seconds since 2000 (local time).

End[s]

Defines end of forensic search in seconds since 2000 (local time).

reserved

This value should be set to zero.

Script length

Length of the script. If the length of the script is zero a default search is performed, i.e. "Detect any object or Detect any Flow or Detect any Motion".

Script mode

The mode defines how the configuration data has to interpreted. See detailed description of configuration data for each case below.

Mode 0x0	Rule engine configuration
Mode 0x1	Rule engine script only
Mode 0x2	In field search
Mode 0x3	Crossing line search

VCA Profile ID

VCA profile id search is supported. The recommended profile id is zero resulting in an exhausting search over the given interval. With a non zero profile id the seacrh speed can be increased because meta data recorded with different a profile are skipped.

Layer Mode

The recommended layer mode value is 0. In this case the layer mask is ignored and the default layers are used. If 1 then the user defined layer mask, see below, is used. The main purpose for the layer mode is for debugging.

Flags

0x01	Flag Send:	Send layer data
0x02	Flag Verbose:	Additional printouts
0x04	Flag Cache Data:	Collects compressed VCD data, next search will use collected VCD packets
0x08	Flag Clear Cache:	Collected VCD packets will be deleted, next search will start new replay

Layer Mask

The layer mask is only used if layer mode is set to 1.

Configuration data

The script mode defines how this data has to be interpreted.

Rule engine configuration

In this mode the configuration data contains the whole rule engine configuration (eg including camera calibration). For more detailed information see the rcp command: CONF_VCD_OPERATOR_PARAMS.

Rule engine script only

If the script mode is selected to "rule engine script only" the configuration data contains the rule engine script. The script is encoded in plain text. One of these examples can be used as a valid script.

Example 1 ("Any motion/flow/object in Field"):

```
Field #1 := { Point(50, 50) Point(100, 50) Point(100, 100) Point(50, 100) };
FlowDetector #1 := { Field #1 };
MotionDetector #1 := { Field #1 };
external SimpleState #1 := DetectedFlow #1;
external SimpleState #2 := DetectedMotion #1;
external ObjectState #3 := InsideField #1;
```

Example 2 ("Object in Field"):

```
Field #1 := { Point(50, 50) Point(100, 50) Point(100, 100) Point(50, 100) };
external ObjectState #1 := InsideField #1;
external Event #2 := OnSet ObjectState #1;
```

Example 3 ("Crossing Line"):

```
Resolution := { Min(-1,-1) Max(1,1) };
Line #1 := { Point(-0.5, -0.5) Point(0.5, 0.5) DebounceTime(0.50) Direction(1) };
external Event#2 := { CrossedLine #1 };
```

Example 4 ("Following Route")

```
Route #2 := { Point(41, 117) Distance(5) Point(51, 87) Distance(9) Point(79, 74)
Distance(5) Direction(1) MinPercentage(80) MaxGap(10) };
external Event #3 := { FollowedRoute #2 };
```

For more details, read the Bosch Query Language Specification.

In field search

A field search is define by two normalized points. Each point is normalized between 0 and 65535. The upper left corner of the image has the coordinates (0,0) and the lower right corner (65535,65535). Additionally, debounce time and intersection mode can be addressed.

16	32
Upper left x-coordinate 2 Byte	**Upper left y-coordinate** 2 Byte
Lower right x-coordinate 2 Byte	**Lower right y-coordinate** 2 Byte
Debounce time 2 Byte	**Field flags** 2 Byte
8	24

Debounce time

The debounce time is given in units of 10 ms.

Field flags

0x01	Object mode:	BaryCenter (0, default) or Boundingbox (1)
0x02	Field mode:	Intersection (o, default) or Covering (1)

Line crossing search

A line crossing search is define by two normalized points. Each point is normalized between 0 and 65535. The upper left corner of the image has the coordinates (0,0) and the lower right corner (65535,65535).

16	32

Upper left x-coordinate 2 Byte	**Upper left y-coordinate** 2 Byte	
Lower right x-coordinate 2 Byte	**Lower right y-coordinate** 2 Byte	
Debounce time 2 Byte	**Direction** 1 Byte	**Reserved** 1 Byte
8	24	

Debounce time

The debounce time is given in units of 10 ms.

Direction

One can choose whether any object which passes the line triggers an event or whether only objects which pass from left to right respectively right to left are relevant:

0 any object which passes the line triggers an event

1 whether only objects which pass from left to right triggers an event

2 whether only objects which pass from right to left triggers an event

CONF_HD_REPLAY_FORENSIC_SEARCH_CANCEL

Tag code	NumDes	Message	SNMP Support
0x0b50	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	-
Write	t_dword	access_right_user	cancel a forensic search, t_dword is the search id (result of search setup), session id is required

CONF_HD_REPLAY_FORENSIC_SEARCH_RESULT

Tag code	NumDes	Message	SNMP Support
0x0b0b	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	results of a forensic search (msg only)
Write	p_octet	access_right_user	results of a forensic search (msg only), see detailed description

This command is a messages command. It successively delivers the results of a forensic search request.

Payload Structure

16 32

Forensic Search ID 4 Byte		
Sequence Number 2 Byte	**Nbr of Evts (n)** 1 Byte	**Last Message** 1 Byte
Progress Time 4 Byte		
Timezone 1 Byte	**Event type** 1 Byte	**Reserved** 2 Byte
n Forensic Search Events n * m byte		

8 24

Forensic Search ID

With the forensic search ID this event message can be assigned to the corresponding search request.

Sequence Number

The sequence number is increasing and missing packets can be detected.

Last Message

The last message byte is signalling that the search has been accomplished with this message.

Progress Time

This time informs about the progress of the search. The time is given in seconds since 1-1-2000 (local time). This time is updated every second (search time) and forces to throw this message. Hence, it can be expected that at least every second a message will be thrown.

Timezone

Timezone in quarter hours (signed byte in two's complement representation).

Event type

Different events can be sent by this message. Error messages include only one event per message.

Object Event	0x00
Error Event	0x01

Object Event

16 32

Begin of event in seconds since 2000 4 Byte		
Residual ms of begin time 2 Byte	**Rule ID** 1 Byte	**Reserved** 1 Byte
Event duration in ms 4 Byte		
Object ID 4 Byte		
Bounding Box 4 Byte		

8 24

Begin

The beginning of the event is provided in local time in seconds since 2000 and milliseconds.

Duration

Duration of the event in ms.

Rule ID

The Rule ID informs which rule is related to this event.

Object ID

Object ID which has caused the alarm.

Bounding Box

Bounding box of object in normalized coordinates from 0 to 255. First, the upper left point is encoded then the lower right point of the bounding box. The coordinate (0,0) is the upper left corner of the image and the point (255,255) is the lower right corner of teh image. The bounding box is from the begin time.

Error Event

Error type	Error text
1 Byte	64 Byte

Error types

Name	Value
e_ForensicSearchErrorNone	0x00
e_ForensicSearchErrorInternal	0x02
e_ForensicSearchErrorNoMetaData	0x07
e_ForensicSearchErrorDataDoesNotMatch	0x08

Error text

The error text is a detailed description of the error. E.g. for syntax errors the line of the script and error type are mentioned.

CONF_HD_REPLAY_PREFETCH_JPEGS

Tag code	NumDes	Message	SNMP Support
0x0b55	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	write only
Write	p_octet	access_right_user	sessionID required, 16bytes header (1byte jpeg quality 0-100, 1byte reserved, 2bytes jpeg height in pixel, 12bytes reserved), N times seconds_since_2000 (each 4bytes)

CONF_HD_REPLAY_CUSTOM_SETTINGS

Tag code	NumDes	Message	SNMP Support
0x0b56	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read replay custom settings, see detailed,
Write	p_octet	access_right_user	set replay custom settings, see detailed,

Payload Structure

16 32

flags 32 Bits
flags mask 32 Bits
kbps max 32 Bits
time shift 32 Bits

8 24

flags

flag field for configuring custom behavior of an replay session

Values:

patch rtp hdr	0x00000001
dont send dummy pkt	0x00000002
h264 only	0x00000004
auto start	0x00000008
no rcp message	0x00000010
set kbps max	0x00000020
L16 to G711	0x00000040
use UTC	0x00000080
add rtp hdr extension	0x00000100
patch pic id	0x00000200
set time shift	0x00000400
transcode	0x00000800
ignore start and end of files	0x00001000
aac 16 khz rtp clock	0x00002000

flags mask

mask for the flag field, it is used to set only choosen values without changing the other values. To set a choosen value in the flag field, the corresponding flag in the flags mask has to be set also. All other setting will be left unchanged

kbps max

set the max send data rate in kbit per seconds, "set kbps max" flag has to be set in order to change that value.

time shift

set time shift in seconds, "set time shift" flag has to be set in order to change that value.

CONF_HD_REPLAY_CUSTOM_SETTINGS

This command is used to configure the behaviour of an existing replay session. A valid replay session id has to be provided.

CONF_HD_REPLAY_CERTIFICATES_LIST

Tag code	NumDes	Message	SNMP Support
0x0c12	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get certificates list of records, Replay Session ID is needed, see detailed description
Write	p_octet	access_right_user	not supported

Request

Payload Structure

16 32

Start Time 4 Bytes	
Stop Time 4 Bytes	
TZ QH 1 Byte	**reserved** 3 Byte
Max List Len 4 Bytes	

8 24

Start Time

Start time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

End Time

End time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Max List Len

Maximum size of the certificates list in response payload in bytes

Reply

Payload Structure

16 32

Start Time 4 Bytes		
Stop Time 4 Bytes		
TZ QH 1 Byte	**Flags** 1 Byte	**reserved** 2 Byte
List Len 4 Bytes		
Certificates List max 16 kb		

8 24

Start Time

Start time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

End Time

End time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Flags

Values:

Bit 0	More certificates available, but the max length of the certificates list was exceeded

List Len

Size of the certificates list in response payload in bytes

Certificates List (sequence of)

16		32
Timestamp 4 Bytes		
TZ QH 1 Byte	**reserved** 1 Byte	**Length** 2 Byte
Certificate n bytes		
8	24	

Timestamp

Local time in seconds since 2000

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Length

Length of the Certificates List Entry (certificate including these 8 bytes header infos)

Certificate

One certificate

CONF_HD_REPLAY_CERTIFICATES_LIST

This command can be used to query the certificates from the records of a camera via replay session (session ID required), in order to verify the signed video record data within that records. The response will deliver all certificates of the records and time intervall addressed by the specidfied replay session. The maximum size of the response can be limited by the caller but it will not exceed 16 kb for the certificates list. If the response payload size isnt't enough to hold all valid certificates, a flag in the response will signal the existence of further certificates, which didn't fit in the response payload. A second query in that case with a smaller time interval may help to get the remaining certificates. The response will have a certificates list with several enries of different size, there are no gaps between the entries.

CONF_HD_REPLAY_SEQUENCE_VERIFY

Tag code	NumDes	Message	SNMP Support
0x0c23	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	verfies authenticity of the replayed video records if enabled, reads the enabled verification flags: 0x1 - hash verify, 0x2 - signature verify, 0x4 - certificate verify
Write	t_dword	access_right_user	verfies authenticity of the replayed video records if enabled, sets the enabled verification flags: 0x1 - hash verify, 0x2 - signature verify, 0x4 - certificate verify

CONF_BACKUP

Tag code	NumDes	Message	SNMP Support
0x0af4	cam	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_user	see detailed description

Payload Structure

16 32

From2000 4 Bytes			
To2000 4 Bytes			
account 4 Bytes			
recordingIndex 4 Bytes			
remoteCamera 1 Byte	**transcode** 1 Byte	**transcoderPreset** 1 Byte	**reserved** 1 Byte
replaySpeed 4 Bytes			
objectID 4 Bytes			
reserved 4 Bytes			
backup filename 16 Bytes			
flags 1 Byte	**signature hash type** 1 Byte	**reserved** 2 Bytes	

8 24

From2000

Start time of the backup in seconds since 2000

To2000

Stop time of the backup in seconds since 2000

account

index of the account that should be used for backup (see CONF_ACCOUNT_SETTINGS), special account dummy backup 0xff.

recordingIndex

index of the recording to be backuped (0 or 1)

remoteCamera

0 - local Camera, 1 to 4 refering to entries of CONF_ADD_REMOTE_DEVICE

transcode

optional field that defines if the video should be transcoded before backup (used by Transcoder Devices only)

transcoderPreset

optional field that defines the preset used by transcoding (used by Transcoder Devices only)

replaySpeed

optional field that defines the replay speed (used by Transcoder Devices only). Use the same values as for CONF_HD_REPLAY_START

objectID

optional field: if that Id is given a autotracker instance will be started that follows the object (used by Transcoder Devices only)

reserved

reserved for future use

filename

backup filename: filename in arbitrary order of: %b begin date/time, %e end date/time, %c camera name, %i export job id, %f file nbr, %a alarm description, %dSubDirectory sub directory to write; optionally seperated by "_" or "-". Exampe: %b_%c would result e.g in a filename 20111017_14-29-26_camera1.mp4

flags

Flags:

CHECK_SIGNATURE	0x01	checks the signature of the source records, the generation of a signature of the backup will fail, if the check fails
AUTH_MSG_ADDON	0x02	Status message CONF_BACKUP_STATUS will contain a message addon containing authenticity check result information

signature hash type

activates the generation of a signature file using the https certificate, works for ftp and dropbox backup

Types:

NO_SIGNATURE	0	signature generation disabled
SHA1	1	generates as signature file over the backup file using SHA1 hash
SHA256	2	generates as signature file over the backup file using SHA256

CONF_BACKUP

starts an mp4 backup file of the specified replay range and store it on a account. The CONF_BACKUP_STATUS message is sent periodicallay to inform about the backup progress. The command returns an 1 byte backup id that can be used to identify the backup in the message. The backup may include signature generation and video authenticity checks. The check results can be return by an optional addon in CONF_BACKUP_STATUS message. The backup destination can be a dummy backup, the data won't be stored anywhere in that case. The dummy backup is used on the special account number 0xff (255). This is usefull, if the backup is just just for authenticity check of records.

CONF_BACKUP_MAX_KBPS

Tag code	NumDes	Message	SNMP Support
0x0af5	account number (1...4) or 0 for http backup	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	gets the maximum datarate for the backup
Write	t_dword	access_right_user	sets the maximum datarate for the backup (in kbps)

CONF_BACKUP_STATUS

	Tag code	NumDes	Message	SNMP Support
	0x0af8	Backup Session ID (if set to 0 a list of all backup sessions is returned)	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	list the status of a backup sessions:	
Write	p_octet	access_right_user	not supported	

This command returns a list of the current backups or the status of one single backup if a specific valid backup id was provided. The list contains the following sequence:

Payload Structure

16 32

Backup ID 4 Bytes			
RCP Session ID 4 Bytes			
Start Time 4 Bytes			
End Time 4 Bytes			
Transfered Bytes 8 Bytes ...			
File Count 16 Bits		**File Nbr** 16 Bits	
File part 16 Bits		**reserved** 16 Bits	
Current time 4 Bytes			
Ppm Current File 4 Bytes			
Backup Errrors 4 Bytes			
RTP packet loss 4 Bytes			
MP4 box errors 4 Bytes			
Account errors 4 Bytes			
Cam 16 Bits		**RecIdx** 16 Bits	
account 1 Bytes	**backup ID** 1 Bytes	**timezone quarter hours** 1 Bytes	**reserved** 1 Bytes
link 128 Bytes			
reserved 8 Bytes			

8 24

Backup ID

Backup Session ID

RCP Session ID

corresponding RCP Session ID

Start Time

Start Time of the Backup in Seconds since 2000

End Time

End Time of the Backup in Seconds since 2000

Current Time

the current Time of the backup

Transfered Bytes

transfered bytes

File Count

number of files to backup

File Nbr

Current file to backup

File part

if a larger Backup file is split into multiple parts this value tells the current part

reserved

reserved

Ppm Current File

status in ppm of the current backup

Backup errors

backup errors: The following errors are defined:

Error during backup	0x02
Backup canceled	0x04
Start of backup failed	0x05

RTP packet loss

number of lost packets

MP4 box errors

errors in the mp4 converter.

Account error

The last Account error. The following errors are defined:

NO_ERROR	0x00
NO_IP	0x01
CONNECT_FAILED	0x02
SETDIR_FAILED	0x03
WRITE_FAILED	0x04

Cam

camera

RecIdx

Recording Index

account

the index of the account used for backup (see CONF_ACCOUNT_SETTINGS and CONF_BACKUP). For backup via http the account is set to 0.

backup ID

the backup id that is returned by the CONF_BACKUP command. This id can be used to link the backup started by CONF_BACKUP with this status message

timezone quarter hours

signed time zone offset in quarter hours

link

a direct link to the backup mp file. The message contains the link if it is available

This command is also send as message by an running backup with same payload and an optional addon containing authenticity informations if the check for it and the addon was activated in the backup job:

Payload Structure MSG

16 32

Backup Status 200 Bytes
Backup Status Auth Addon (optional) 24 Bytes

8 24

Backup Status Addon (optional)

16 32

start sec utc 4 Bytes			
end sec utc 4 Bytes			
TZ QH 1 Bytes	**hash status** 1 Bytes	**sign status** 1 Bytes	**cert status** 1 Bytes

hash issues 4 Bytes
sign issues 4 Bytes
cert issues 4 Bytes

8 24

start sec utc

start time in seconds since 2000 utc of the sequence

end sec utc

last time in seconds since 2000 utc of the sequence

TZ QH

time time zone offset in quarter hours (7 lowest bits), and sign (highest bit)

hash status

hash status for authenticity check

Values:

missing	0
not checked	1
invalid	2
valid	3

sign status

signature status

Values:

missing	0
not checked	1
missing certificate	2
invalid	3
valid	4

cert status

signing certificate status

Values:

not checked	0
unknown	1
invalid	2
trusted	3
owned	4

hash issues

count of hash issues occured since start of authenticity check

sign issues

count of sign issues occured since start of authenticity check

cert issues

count of cert issues occured since start of authenticity check

CONF_BACKUP_STOP

	Tag code	NumDes	Message	SNMP Support
	0x0b58	yes, Backup Session ID	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	not supported	
Write	t_dword	access_right_user	cancels an ongoing backup job	

CONF_FORMAT_FS

	Tag code	NumDes	Message	SNMP Support
	0x09d0	storage medium (0=DEFAULT is recording storage device, 1=recording storage device, 2=RAMDISK)	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_service	obsolete	
Write	t_dword	access_right_service	obsolete	

CONF_FORMAT_FS_STATUS

Tag code	NumDes	Message	SNMP Support
0x09d1	yes (file system nr, 1 - recording storage device, 2 - RAMDISK)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	status and progress of a running format requested by CONF_FORMAT_FS
Write	p_octet	access_right_minimal	status and progress of a running format requested by CONF_FORMAT_FS, see detailed description

Payload Structure

8	16	32
step 1 Byte	**status** 1 Byte	**progress** 2 Byte

step

Which step of the formating (FAT32 Formatting including creating recording files or copy replay tools)

Values:

HD_STEP_FORMAT_FS	0x00
HD_STEP_COPY_TOOLKIT	0x01

status

reports the state of formatting. The format can be in state of running, failed or successfully finished. The last one is devided in the two states of HD_FORMAT_STATUS_FINISHED which means FAT32 including recording files and replay tools were created and HD_FORMAT_STATUS_SKIPPED which means that only the FAT32 was created because the storage device was not the needed size for recording.

Values:

HD_FORMAT_STATUS_FAILED	0xff
HD_FORMAT_STATUS_RUNNING	0x00
HD_FORMAT_STATUS_SKIPPED	0x01
HD_FORMAT_STATUS_FINISHED	0x02

progress

value from 0 to 100 represent the progress of the format including copy replay tools in percent.

Values:

progress in percent	0 to 100

CONF_FORMAT_FS_STATUS

the message reports the status and progress of a running format. The formatting takes two steps that is first formating FAT32 including creating the recording files and second copy the replay tools. The progress is reported about the whole formatting. The status reports the state of each formatting step. A successful format should always end with a message with 100 percent progress, step should be copy toolkit and status should be HD_FORMAT_STATUS_FINISHED or HD_FORMAT_STATUS_SKIPPED if the storage device is too small for recording (e.g. RAMDISK). A failed format should finish with a status HD_FORMAT_STATUS_FAILED at any progress and step. After this final messages no futher message will follow.

CONF_FORMAT_FS_SPAN

Tag code	NumDes	Message	SNMP Support
0x09e6	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	format the lun into spans.
Write	p_octet	access_right_service	format the lun into spans. (not allowed while recording on this lun)

Payload Structure

16 32

Server IP 4 Bytes		
Target IDX 1 Byte	**Lun IDX** 1 Byte	**Span Cnt** 2 Byte
Span Size 4 Bytes		
Flags(optional) 1 Byte	**reserved** 3 Byte	

8 24

Server IP

The iSCSI server ip address

Values:

Local connected USB disk	ff.ff.ff.ff
iSCSI Server	valid ip address

Target IDX

Index of the iSCSI server target. The index of the desired target can be obtained from the reply to an iSCSI discover.

Lun IDX

Index of the lun of the specified target.

Span Cnt

Number of spans the lun shall be formated with

Span Size

Size of each span in megabyte. (1 MB = 1024 x 1024 Bytes)

Flags

these flags are optional, if no flags are set (default), no span format will be executed

REPAIR_FS (repair file system without clearing spans)	0x01
CLEAR_SPANS_ONLY (only clear the spans)	0x02

Note

The reply packet will contain the values of the actually formated spans and size. If the product of the span count and the span size in the request packet exceeds the size of the current lun, only as many as possible spans are formatted. If the product is less than the available lun space, the back part of the lun will be unused. Size of the write payload is at least 12 bytes, if the optional flags are uses the payload has to be at least 16 bytes.

Specific Error Types

SPAN_ERR_INTERNAL	0x01
SPAN_ERR_NOT_MNTD	0x05
SPAN_ERR_INV_FS	0x06
SPAN_ERR_INV_LUN_NFO	0x07
SPAN_ERR_RD_ONLY	0x0a
SPAN_ERROR_OLD_LUN_NFO	0x20
ISCSI_ERROR_CONNECT	0x31
ISCSI_ERROR_INV_LUN	0x33
ISCSI_ERROR_LOGIN	0x34
ISCSI_ERROR_INV_TRG_IDX	0x35

Note: A list of all defined error types can be found in the Appendix.

CONF_FORMAT_FS_SPANS_STATUS

Tag code	NumDes	Message	SNMP Support
0x09f5	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	status and progress of a running format requested by CONF_FORMAT_FS_SPANS, see detailed description
Write	p_octet	access_right_service	status and progress of a running format requested by CONF_FORMAT_FS_SPANS, see detailed description

Payload Structure

16 32

step 1 Byte	**status** 1 Byte	**progress** 2 Byte
ip 4 Bytes		
target_idx 1 Byte	**lun** 1 Byte	**reserved** 2 Byte

8 24

step

Which step of the formating (FAT32 Formatting including creating spans and clearing span headers)

Values:

HD_STEP_FORMAT_FS 0x00

HD_STEP_CLEAR_SPAN_HEADER 0x02

status

reports the state of formatting. The format can be in state of running, failed or successfully finished. success means the creation of the fat32 file system and the empty spans or clearing the span headers was successfully.

Values:

HD_FORMAT_STATUS_FAILED 0xff

HD_FORMAT_STATUS_RUNNING 0x00

HD_FORMAT_STATUS_FINISHED 0x02

HD_FORMAT_ERASE_DISK 0x03

progress

value from 0 to 100 represent the progress of the format including the creation of the empty recording spans and clearing the header files.

Values:

progress in percent 0 to 100

ip

ip of the storage device or lun, that is formatting

target_idx

Target Index of the storage device from 0 to 255.

lun

Lun of the storage device from 0 to max. 255.

reserved

2 Bytes reserved

CONF_FORMAT_FS_SPANS_STATUS

this message reports the status and progress of a running format. The formatting includes the creation of the FAT32 file system (step: HD_STEP_FORMAT_FS) and clearing of the span header files(step: HD_STEP_CLEAR_SPAN_HEADER). The progress is reported about the whole formatting. A successful format should always end with a message with 100 percent progress at step HD_STEP_CLEAR_SPAN_HEADER and status should be HD_FORMAT_STATUS_FINISHED. A failed format should finish with a status HD_FORMAT_STATUS_FAILED at any progress and any step. After these final messages no futher messages will follow. The whole formatting takes two steps HD_STEP_FORMAT_FS and HD_STEP_CLEAR_SPAN_HEADER. First step HD_STEP_FORMAT_FS will run up to 90 percent and will end with a 90 percent progress message and status HD_FORMAT_STATUS_FINISHED if successful. Then the last step HD_STEP_CLEAR_SPAN_HEADER will follow from 90 to 100 percent and will also end with a HD_FORMAT_STATUS_FINISHED status if successfull, which is also as mentioned earlier the final success message. If the formatting runs with the option of disk erasing, The first step will be the HD_STEP_ERASE_DISK step and will run upto 70 percent if successful.

CONF_SPAN_ADDRESS_LIST

Tag code	NumDes	Message	SNMP Support
0x09e7	yes (1 - primary spanlist, 2 - secondary spanlist)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	See detailed description.
Write	p_octet	access_right_service	See detailed description.

Payload Structure

Span Address 1 8 Bytes ...	
(n-2) x 8 Bytes	**Span Address n** 8 Bytes ...

Span Address 1 - n

For the structure of a span address so description below.

16 32

Targt ID 4 Bytes		
Target IDX 1 Byte	**Lun IDX** 1 Byte	**Span IDX** 2 Byte

8 24

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

Index of the iSCSI server target. The index of the desired target can be obtained from the reply to an iSCSI discover.

Lun IDX

Index of the lun of the specified target.

Span IDX

Index of the span in the specified lun.

Write request

On a write request, a list of max n (n=512) span addresses must be supplied. List entrys that are zero are not used by the unit for recording. The unit uses the supplied ordering of the list and begins with the first entry. The order of the supplied list is the preferred order, not the guaranted order. There are some cases, in which the device changes the use order (e.g. one target not reachable).

If the device is already recording on spans (a span list is already present on the device), the new list must contain the span addresses, that the unit is currently recording on. The latter information can be obtained by reading the current span address list form the unit and check the corresponding write lock header (if the units IP and MAC address is found this means the unit is recording). If the unit receives a list and the span addresses of the spans the device is currently recording on is not present, the list is rejected (not saved in the configuration) and an error is returned

Read Request

The reply to an read request always contains up to n (n=512) entrys. The response in case of an empty span list is one zeroed span entry.

Message

A message of this cmd tag will be send always, if somone sets the span address lists by using rcp set spans list commands or if the span list is cleared by indirectly by changing other settings of the device (e.g. rec mgmt).

Specific Error Types

SPAN_ERR_INTERNAL	0x01
SPAN_ERR_INV_ADDR_LIST	0x04

Note: A list of all defined error types can be found in the Appendix.

CONF_SPAN_ADDRESS_LIST_NEW

Tag code	NumDes	Message	SNMP Support
0x0a52	yes (1 - primary spanlist, 2 - secondary spanlist)	yes	no

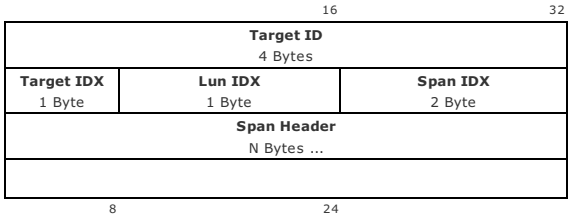
	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	See detailed description of CONF_SPAN_ADDRESS_LIST
Write	p_octet	access_right_service	See detailed description of CONF_SPAN_ADDRESS_LIST (no need to include mounted spans in list)

CONF_SPAN_HDR_ACCESS

Tag code	NumDes	Message	SNMP Support
0x09e8	yes	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	Access span header information.
Write	p_octet	access_right_service	Access span header information.

Payload Structure



Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Values:

Local connected USB disk	ff.ff.ff.ff
iSCSI Server	Valid IP address

Target IDX

Index of the iSCSI server target. The index of the desired target can be obtained from the reply to an iSCSI discover.

Lun IDX

Index of the lun of the specified target.

Span IDX

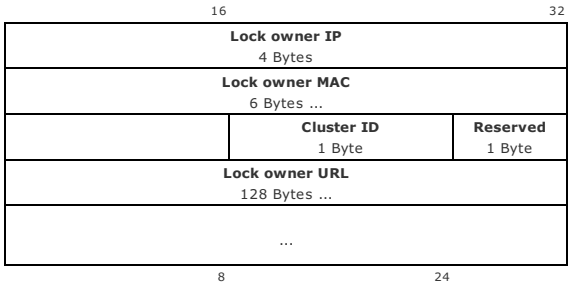
Index of the span in the specified lun.

Span Header

Data of the span header. Length and structure differs for the specified header type. The header type is specified with the **numeric descriptor**:

Values:

LOCK HEADER	0x01



Lock owner IP

IP address of the unit that holds the write lock of the span.

Lock owner MAC

Hardware address of the unit that holds the write lock of the span

Cluster Id

Cluster id of the recorder 1 - n, default value is 0 if not set, should be treated as cluster id 1

Lock owner URL

url (ipv4 or ipv6) of the unit that holds the write lock of the span

NOTE:

In the request packet, the header values are ignored (may be zero). The reply contains the values read or written.

On write requests, the local IPv4 and MAC addresses are converted to ascii strings

and written to the file LCKxxxxx.txt (xxxx = span index) on disk. On read requests, the ascii strings of that file are scanned into binary values and send with the reply.

MANAGER HEADER 0x02

16	32
Manager Header Data N Bytes (max. 1024) ...	
8	24

Manager Header Data

Transparent header data of the storage manager

The reply always returns the 1024 Bytes of the manager header file MGRxxxx.txt (xxxx = span index).

UNIT HEADER 0x03

16			32
Span owner IP 4 Bytes			
Span owner MAC 6 Bytes ...			
		Span owner Camera 2 Byte	
Retention Time 4 Bytes			
Modification seal random 4 Bytes			
Modification seal time 4 Bytes			
Recording 1 Byte	**Cluster ID** 1 Byte	**encryption** 1 Byte	**reserved** 1 Byte
Max Retention Time 4 Bytes			
Span owner URL 128 Bytes ...			
...			
Modification MAC 6 Bytes ...			
		reserved 2 Byte	
User Data 16 Bytes ...			
...			
GUID 32 Bytes ...			
...			
8		24	

Span owner IP

IP address of the unit that holds the write lock of the span.

Span owner MAC

Hardware address of the unit that holds the write lock of the span

Span owner Camera

The index of the camera the unit uses for recording.

Retention Time

Retention Time for the recordings on this span in sec since 2000

Modification seal random

random number set on the last modifacation of this span, this random will be set on each write access on any span header or on starting or stopping a recording of a device on this span, there will be also updates while a device is recording in a period of several minutes

Modification seal time

this information is the time of the last modification on this span that causes a modification seal update

Recording

for dual recording, 1 - primary recording, 2 - secondary recording, if this value is not set then default is primary recording

Cluster Id

Cluster id of the recorder 1 - n, default value is 0 if not set, should be treated as cluster id 1

encryption

encryption of the span, 0 - unknown (check record file for keys, will be ignored for write direction), 1 - no encryption, 2 - xts encrypted

Max Retention Time

Max Retention Time for the recordings on this span in sec since 2000, 0 means no max retention time

Span owner URL

url (ipv4 or ipv6) of the unit that used the span

Modification MAC

Hardware address of the instance, which modifies the unit header (must be non zero for write direction)

User Data

byte field for any user data (16 bytes)

GUID

GUID (managed by vrm) of the device which recorded on this span (32 bytes)

NOTE:

In the read request packet, the header values are ignored (may be zero). In the write requst packet, only the camera field is evaluated, the values for IP and MAC are taken from local configuration. The reply packet contains the values read or written.

On write requests, the local IPv4 and MAC addresses and the supplied camera value are converted to ascii strings and written to the file UNTxxxxx.txt (xxxx = span index) on disk. On read requests, the ascii strings of that file are scanned into binary values and send with the reply.

The modification seal values cannot be set by this command as write request directly, they will be set automatically on span header write requests or on starting or stopping a recording. The content of this fields will be ignored in a write request.

Specific Error Types:

SPAN_ERR_INTERNAL	0x01
SPAN_ERR_INV_SPN_IDX	0x02
SPAN_ERR_INV_HDR_TYPE	0x03
SPAN_ERR_NOT_MNTD	0x05
SPAN_ERR_INV_FS	0x06
SPAN_ERR_INV_LUN_NFO	0x07
SPAN_ERR_BAD_HDR_CKSM	0x08
SPAN_ERR_RD_ONLY	0x0a
ISCSI_ERROR_CONNECT	0x31
ISCSI_ERROR_INV_LUN	0x33
ISCSI_ERROR_LOGIN	0x34
ISCSI_ERROR_INV_TRG_IDX	0x35

Note: A list of all defined error types can be found in the Appendix.

CONF_START_SPAN_RECORD

Tag code	NumDes	Message	SNMP Support
0x09f7	yes (1 - primary recording, 2 - secondary recording)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	obsolete, please use CONF_START_RECORD instead
Write	p_octet	access_right_service	obsolete, please use CONF_START_RECORD instead

CONF_SPAN_PARTITION_PROP

Tag code	NumDes	Message	SNMP Support
0x09fa	yes (camera)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	return partition properties for primary span recording (4 bytes) video type (1 byte: 0= NO, 1=MPEG2, 3=MPEG4), audio type (1 byte: 0= NO, 1=G711, 2=L16, 3=AAC) reserved (2 byte)
Write	p_octet	access_right_service	(obsolete and not supported anymore)

CONF_SPAN_PARTITION_PROP_ALARM

Tag code	NumDes	Message	SNMP Support
0x09fb	yes (camera)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	obsolete and not supported anymore
Write	p_octet	access_right_service	obsolete and not supported anymore

CONF_SPAN_PARTITION_PROP_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4b	yes (camera)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	return partition properties for secondary span recording (4 bytes) video type (1 byte: 0= NO, 1=MPEG2, 3=MPEG4), reserved (3 byte)
Write	p_octet	access_right_service	(obsolete and not supported anymore)

CONF_SPAN_PARTITION_PROP_ALARM_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4c	yes (camera)	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	obsolete and not supporded anymore
Write	p_octet	access_right_service	obsolete and not supported anymore

CONF_SPAN_USE_STATUS

Tag code	NumDes	Message	SNMP Support
0x09f8	cam	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	usage status of a span, see detailed description
Write	p_octet	access_right_service	not supported

Payload Structure

16	32
use status message 1 12 Bytes ...	
use status message 2 12 Bytes ...	

8 24

use status message 1 - 2

Upto 2 span use status messages. If no span is in use by the camera specified by num, the reply payload size will be 0. In case of the rcp message the payload contains always 1 span use status message.

Span Use Status Message Structure

	16	32
err/spec code 1 Byte	**status** 1 Byte	**cam** 2 Bytes
target_id 4 Bytes		
target_idx 1 Byte	**lun** 1 Byte	**span_idx** 2 Byte

8 24

error/special code

additional error or status code to the status, for status HD_SPAN_USE_STATUS_ERROR_MOUNT see section "status HD_SPAN_USE_STATUS_ERROR_MOUNT error codes", for status RELEASE and RELEASE_UNUSED see section "status RELEASE error codes", for status MOUNTED see section "status MOUNTED special codes"

status

use status of this span

Values:

MOUNTED	0x00
RELEASE	0x01
RELEASE_UNUSED	0x02
ERROR	0x03
RECORDING	0x04
PENDING_SPAN_REQUEST	0x05
HD_SPAN_USE_STATUS_RETENTION_TIME	0x06
HD_SPAN_USE_STATUS_RELEASE_REMOUNT_INTENDED	0x07
HD_SPAN_USE_STATUS_ERROR_WRT_UNT_HDR	0x10
HD_SPAN_USE_STATUS_ERROR_FORMAT_REC_REGION	0x11
HD_SPAN_USE_STATUS_ERROR_MOUNT	0x12
HD_SPAN_USE_STATUS_ERROR_RETENTION_TIME	0x13

If the status is PENDING_SPAN_REQUEST, the ip will be 255.255.255.255, target index and lun will be 255, and span index will be 65535, which should be interpreted as still unknown, because this cam had requested a span, but it doesn't know yet which span it gets.

Status HD_SPAN_USE_STATUS_RELEASE_REMOUNT_INTENDED means, the cam has released the block but has left the span lock, because it will try to remount the span later. The span is stored in the span history for remount(see CONF_SPAN_HISTORY).

cam

which cammera is or was using the reported span

target_id

target id of the lun (may be the ipv4 address for older versions or with default target resolve

configuration, see CONF_TARGET_ID_RESOLVE_RULES)

target_idx

Target Index of the storage device from 0 to 255.

lun

Lun of the storage device from 0 to max. 255.

span_idx

Index of the span on the storage device.

CONF_SPAN_USE_STATUS

This messages reports the use status of a span for this device and which cammera for primary recording. If a span was opened for a planned recording, this message will be send with status MOUNTED. If the mounting fails status will be ERROR. If a span was closed after recording status will be RELEASE, if it was closed before a recording ever took place, status will be RELEASE_UNUSED. HD_SPAN_USE_STATUS_RETENTION_TIME status will be send as message if the first time an retention time update took place on the span. In case of a read request the reply payload cantains this status message upto twice. the command can be used to query the actual used spans from a camera specified by the num parameter. the status for a span can be RECORDING, if the cam is actual recording on this span or MOUNTED, if the span is mounted and prepared for a future recording by this camera.

status HD_SPAN_USE_STATUS_ERROR_MOUNT error codes

Values:

none	0x00
SPAN_ERROR_INTERNAL	0x01
SPAN_ERROR_INV_SPN_IDX	0x02
SPAN_ERROR_INV_HDR_TYPE	0x03
SPAN_ERROR_INV_ADDR_LIST	0x04
SPAN_ERROR_NOT_MNTD	0x05
SPAN_ERROR_INV_FS	0x06
SPAN_ERROR_INV_LUN_NFO	0x07
SPAN_ERROR_BAD_HDR_CKSM	0x08
SPAN_ERROR_INV_IDX	0x09
SPAN_ERROR_RD_ONLY	0x0a
SPAN_ERROR_NO_REC_DAT	0x0b
SPAN_ERROR_INV_PART_NFO	0x0c
SPAN_ERROR_SPAN_REQUEST_FAILED	0x0e
SPAN_ERROR_SPAN_REQUEST_RETENTION_TIME	0x0f
SPAN_ERROR_REMOUNT_REFUSED	0x10
ISCSI_ERROR_CONNECT	0x31
ISCSI_ERROR_INV_LUN	0x33
ISCSI_ERROR_LOGIN	0x34
ISCSI_ERR_PWD	0x36
ISCSI_ERR_PROTO	0x37
ISCSI_ERR_TARG_NOT_REACH	0x38
ISCSI_ERR_NO_MEM	0x3a
ISCSI_ERR_SESS_CREATE	0x3b
ISCSI_ERR_INV_PARAMS	0x3c
ISCSI_ERR_SESS_NOT_FOUND	0x3d
ISCSI_ERR_DISCONN	0x3e
ISCSI_ERR_TIMEOUT	0x3f
ISCSI_ERR_SOCK	0x5f
ISCSI_ERR_TARG_PM	0x6f
ISCSI_SOCK_CLOSED	0x7f
ISCSI_ERR_TCP_CONN_RST	0x8f
ISCSI_ERR_INTR_NOT_SUPP	0x9f
ISCSI_ERR_IP_ZERO	0xa0
ISCSI_ERR_OUT_OF_RES	0xa1
HDD_ERROR_TIMEOUT	0x12
HDD_ERROR_CREATE_FAILED	0x22
HDD_ERROR_ACCESS_DENIED	0x32
HDD_ERROR_DEVICE_PRESENT_TIMEOUT	0x42
HD_PMM_ERROR_LUN_LOCK	0x52
HD_PMM_ERROR_INVALID_ACCESS	0x62
HD_PMM_ERROR_LUN_MGMT_FILE_NOT_FOUND	0x72

HD_PMM_ERROR_LUN_WRITE_PROTECTED	0x82
HD_PMM_ERROR_COMMON	0xf2

status RELEASE error codes

Values:

none		0x00
RELEASE_ERROR_HD_MGR_ERROR		0x01
RELEASE_ERROR_SPAN_LIST_INCONSISTENCY		0x02
RELEASE_ERROR_RECORD_HANDLE_DISMISS_ERROR		0x03
RELEASE_ERROR_BIG_TIME_JUMP		0x04
RELEASE_ERROR_CLOSED_WHILE_MOUNTING		0x05
RELEASE_ERROR_INITIAL_MOUNT_ABORT		0x06
RELEASE_ERROR_INITIAL_MOUNT_UNFINISHED		0x07
RELEASE_ERROR_NEXT_SPAN_NOT_MOUNTED		0x08
RELEASE_ERROR_RECORDING_ERROR		0x09
RELEASE_ERROR_ENCODER_ERROR		0x0a
RELEASE_ERROR_REPLACE	0x0b	(release was caused by CONF_SPAN_SWITCH command)
RELEASE_ERROR_SWITCH_REQUEST	0x0c	(release was caused by CONF_SPAN_SWITCH command)
RELEASE_ERROR_SPAN_TIME_LIMIT_REACHED	0x0d	(one day limit per span, if max retention configured)

status MOUNTED special codes

Values:

standard mount	0x00
HD_MOUNT_CODE_REMOUNTED	0x01

CONF_SPAN_USE_STATUS_SECONDARY

Tag code	NumDes	Message	SNMP Support
0x0a4a	cam	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	span use status for secondary recording, payload as in CONF_SPAN_USE_STATUS
Write	p_octet	access_right_service	not supported

CONF_SPAN_PARTITION_FILE_INFO

Tag code	NumDes	Message	SNMP Support
0x0a2d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Read the file info of a span.
Write	void	access_right_service	not supported

Payload Structure

16 32

Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Span IDX** 2 Byte
Start Time (optional) 4 Bytes		
Stop Time (optional) 4 Bytes		
maxEntries (optional) 4 Bytes		
optional flags (optional) 4 Bytes		

8 24

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

Index of the iSCSI server target. The index of the desired target can be obtained from the reply to an iSCSI discover.

LUN

The logical unit.

Span IDX

Index of the span in the specified lun.

Start Time

Seconds since 2000, optional, if missing, default is 0

Stop Time

Seconds since 2000, optional, if missing, default is unlimited

maxEntries

Max Number of entries, optional, if missing, default is max. 256 files which is also the max. limit for this value

optional flags

additional options for the file info

Values:

add span info flags 0x00000001

Response Structure

For the payload structure of the response see documentation of HD_PARTITION_FILE_INFO. In case of the optional "add span info flag" flag the first four bytes of the responspayload will contain a flag field with additional infos. Without the option, the payload will start with the file info without offset.

Response Payload Structure

optional span info flags 4 Bytes
file info payload n Bytes
...

8 24

optional span info flags

additional info of the span

Values:

encrypted span key list

present	0x00000001

CONF_EXPORT_SPAN

Tag code	NumDes	Message	SNMP Support
0x0a2e	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Retrieve a span of the devices storage.
Write	p_octet	access_right_service	not supported

Request Payload Structure

Recording Camera 12 Bytes ...
Lun Addr 8 Bytes ...
Reserved 12 Bytes ...
Overwrite Retention Time 4 Bytes
Reserved 4 Bytes
Url max. 128 Bytes

...

Recording Camera

The address of the device the span is assigned to.

16 32

IPv4 4 Bytes		
MAC 6 Bytes ...		
	Recording Index 1 Byte	**Camera** 1 Byte

8 24

IPv4

The ipv4 address of the recording device. If the device hasn't an ipv4 Adress, this field should be set to invalid (zero ip 0.0.0.0). In that case a valid url has to be provided.

MAC

The hardware address of the recording device.

Recording Index

1 = Primary Recording, 2 = Secondary Recording

Camera

The camera index.

Lun Addr

The address of the lun a span is desired from.

16 32

Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Reserved** 2 Byte

8 24

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

The index of the target.

LUN

The logical unit number.

Overwrite Retention Time

If this field is set (> 0) and no free spans are available, spans with retention times that are lower or equal to the specified retention time are accounted for export. In this case the recording with the retention time that expires next is overwritten.

Url

Url of the the device which requests the span as zero terminated ascii string. Max length is 128 bytes including zero termination. This parameter is optional but is requiered if the device doesn't has an ipv4 address or if the IPv4 adress of the Cam id is not used.

Response and Message Payload Structure

16	32
Recording Camera/Storage Managing Host 12 Bytes ...	
Span Address 8 Bytes ...	
Last Recording 16 Bytes ...	
State 1 Byte	**Reserved** 3 Bytes
8	24

Recording Camera

The address of the device the span is assigned to, only in message payload(see request payload description for details).

Storage Managing Host

The address of the device the span was handed out, only in reply payload(hint: in older versions of that command, the recording camera info was send instead).

16	32
reserved 4 Bytes	
MAC 6 Bytes ...	
	reserved 2 Bytes
8	24

MAC

The hardware address of the device, which handed out the span.

Span Address

The address of the span the device exports. If this field is zero, no span is exported.

16		32
Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Span IDX** 2 Byte
8	24	

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

The index of the target.

LUN

The logical unit number.

Span IDX

The index of the span on the lun.

Last Recording

The address of the span the device exports.

16	32
Recording Camera 12 Bytes ...	
Retention Time 4 Bytes	
8	24

Recording Camera

The id of the camera of the recording that is overwritten. Field is zero, if there was no recording on this span. See request packet for detailed description.

Retention Time

The retention time (seconds since 2000) of the recording that is overwritten.

State

Values:

SUCCESS	0x00
NO SPAN AVAILABLE	0x01
STORAGE OFFLINE	0x02
INVALID SPAN MANAGER ADDRESS	0x03
ACCESS DENIED	0x04

A list of span manager the recording is using can be obtained by the command CONF_REC_SPAN_MGR.

CONF_CAM_REC_SPANS

Tag code	NumDes	Message	SNMP Support
0x0a8f	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	Retrieve the list of recording spans of a camera from a span manager on a managed lun.
Write	p_octet	access_right_service	not supported.

Request Payload Structure

16	32
Recording Camera 12 Bytes ...	
Lun Address 8 Bytes ...	
8	24

Recording Camera

The address of the camera for which recording spans are to be queried.

16		32
IP 4 Bytes		
MAC 6 Bytes ...		
	Recording Index 1 Byte	**Camera** 1 Byte
8	24	

IP

The ip address of the recording device.

MAC

The hardware address of the recording device. This field may be zero if you search for recordings without considering the mac.

Recording Index

1 = Primary Recording, 2 = Secondary Recording

Camera

The camera index.

Lun Address

The address of the lun on which recording spans for a camera are searched.

16		32
Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Reserved** 2 Byte
8	24	

Target ID

target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

The index of the target.

LUN

The logical unit number.

Response and Message Payload Structure

Recording Camera 12 Bytes ...

Lun Address 8 Bytes ...
N x Span Recording Info 16 Bytes ...

Recording Camera

The address of the recording camera (see request payload description for details).

Lun Address

The address of the lun of the recording spans (see request payload description for details).

Span Recording Info

Recording info about the spans the camera recorded.

16			32
Span Index 2 Byte	**S** 1	**Timezone** 7 Bits	**Flags** 1 Byte
Start Time 4 Bytes			
Stop Time 4 Bytes			
Reserved 2 Bytes	**File Count** 2 Bytes		
8	24		

Span Index

The span index in the current lun.

S

The timezone sign.

Timezone

Timezone in quarter hours.

Flags

Values:

Bit 0	Continous Recording
Bit 1	Recording Running
Bit 2	Migrated Recording

Start Time

Seconds since 2000.

Stop Time

Seconds since 2000.

File Count

The number of files that are stored in this span.

CONF_REC_SPAN_MGR

	Tag code	NumDes	Message	SNMP Support
	0x0a36	yes (1 - primary recording, 2 - secondary recording)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal		
Write	p_octet	access_right_service		

Payload Structure

Sequence of (max 8):

16 32

Span Mananger IP address 4 Bytes		
Target ID 4 Bytes		
Target Index 1 Byte	**Lun** 1 Byte	**Reserved** 2 Byte
Priority 2 Bytes		**Flags** 2 Byte

8 24

Span Mananger IP address

The IP address of the device the span manager is running on.

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target Index

The iSCSI target index.

Lun

The iSCSI lun of the storage.

Priority

Priority as recording storage, higher values mean lower priority, 0 - highest priority

Flags

Values:

OVERWRITE_OLDEST_RECORDINGS(obsolete) 0x0001

If no free spans are available on all span manager of the devices recording span manager list, the recording with the retention time that expires next is overwritten.

Use this command to edit the recording span manager list. This list is consulted every time when a device is recording in VRM mode and its span list runs out resources. The entry with the highest priority is picked out of this list, the corresponding span manager is contacted and queried for spans. If the span manager ís not accessable or will not return a span for some reason (maybe it has no more free spans to export), the entry with the next lower priority is picked out of the list. If at least one span manager for primary recording is configured, all entries of managed vrm (configured by CONF_MANAGING_VRM) will be deleted.

CONF_SPAN_REC_COPY

Tag code	NumDes	Message	SNMP Support
0x0a55	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	obsolete
Write	p_octet	access_right_service	obsolete

CONF_SPAN_SWITCH

	Tag code	NumDes	Message	SNMP Support
	0x0a53	yes (cam)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	not supported	
Write	p_octet	access_right_service	switch spans, see detailed description	

Payload Structure

16 32

target id recording span 4 Bytes			
target_idx 1 Byte	**lun** 1 Byte	**span_idx** 2 Byte	
target id premounted span 4 Bytes			
target_idx 1 Byte	**lun** 1 Byte	**span_idx** 2 Byte	
flags 16 Bits		**recording** 1 Byte	**reserved** 1 Byte

8 24

target id recording span

the target id of the actual recording span

target_idx

the target index of the actual recording span

lun

the lun of the actual recording span

span_idx

the span index of the actual recording span

target id premounted span

the target id of the premounted span

target_idx

the target index of the premounted span

lun

the lun of the premounted span

span_idx

the span index of the premounted span

flags

these flags are used to specify the behav. of the command

Flags:

FORCE	0x0001
CHECK_PREMOUNTED	0x0002
REPLACE_PREMOUNTED	0x0004
SWITCH_SPAN	0x0008
TARGET_RETREAT	0x0010
TARGET_REINTRODUCE	0x0020
ERROR_CHECK_PREMOUNTED	0x4000
ERROR_CHECK_RECORDING	0x8000

recording

select a recording

Values:

primary recording	1
secondary recording	2

CONF_SPAN_SWITCH

This command can be used to influence the span usage of the recording. It is able to cause a replace of the premounted span and or switch to the next premounted span.

The flag SWITCH_SPAN causes the switch to the next span and releasing the actual recording

span. Flag REPLACE_PREMOUNTED will cause the release of the premounted span and mounting another one if available. If both flags are activated, the premounted block will be replaced, as soon as a new premounted span is mounted and available, the next step switch to premounted span will be performed. Before replacing the premounted span, the actual recording span will be checked, if the replace could get in conflict with a device internal triggered span switch. In that case the whole Span switch command will fail with rcp error RCP_ERROR_TRY_LATER, no switch or replace will be performed.

When performing a switch to next span job, the recording span address in the payload will be checked. If on execution of this job the recording span doesn't match the recording span from the payload, the switch will not be performed. When performing a replace premounted span job, the premounted span address in the payload will be checked. If on execution of this job the premounted span doesn't match the premounted span from the payload, the replace will not be performed. The idea of these checks is to avoid conflicts with the automatic span switching of the recording, that can lead to gaps in the recording and unwanted waste of recording spans. The flag FORCE will skip these checks. Flag CHECK_PREMOUNTED is only relevant for the switch to next span job without replacing the premounted span and without FORCE flag. If this flag is active on switching ton next span, the premounted span parameter in the payload will be matched against the actual premounted span. If no match the switch won't be performed. If this flag including the FORCE flag isn't set on switch, a negative match will be performed on the premounted span. The TARGET_RETREAT Bit means, than a complete retreate from this target is intended, the span history will be modified in order to avoid a remount on that target. That state will stay active for that target until the device records on a new target and adds a span from the new target to the history, or if the span history will be cleared or the the target will be reintroduced with the bit TARGET_REINTRODUCE. The TARGET_REINTRODUCE Bit is needed to undo a TARGET_RETREAT action, which is specialy in the case needed, when a device shall return to a target, from that it had retreated erlier and no recording took place on another target between these events. In that case the remount feature for that target is still disabled. The TARGET_REINTRODUCE Bit will reenable the remount feature for that target without the nessessarity of clearing the whole span history.

Warning: A possible conflict, when requesting a replace of the premounted span with an automatic span switch of the recording that causes gaps in the recording is unavoidable. So an excesive usage of replacing premounted spans should be avoided.

CONF_HD_SET_VRM_LOCK

Tag code	NumDes	Message	SNMP Support
0x0a5d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_service	Write or clear the vrm lock.

Payload Structure

16 32

Target ID 4 Bytes		
target idx 1 Byte	**lun** 1 Byte	**reserved** 2 Byte
flags 32 Bits		

8 24

Target ID

Target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

target idx

Target index of the storage device

lun

Lun of the storeage device

flags

flags

Flags:

SET_VRM_LOCK 0x01

CONF_HD_SET_VRM_LOCK

This command sets or clears the vrm lock on a lun. If the flag SET_VRM_LOCK is set the vrm lock will be written, otherwise it will be cleared.

CONF_SPAN_FILES_DIR

	Tag code	NumDes	Message	SNMP Support
	0x0aa1	yes (entry index 1 - 32768)	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	Read the path entry end exported iscsi addr. (gen dll only)	
Write	p_octet	access_right_service	Set path to span files (gen dll only).	

Payload Structure (write/in)

Action 4 Bytes		
reserved 8 Bytes ...		
Path name 1 1 Byte	N x 1 Byte	**Path name N** 1 Byte

Action

specifies the action, you can set an entry or clear an entry

Values:

clear entry	0
set entry	1

Path name 1 - N

zero terminated ascii string containing the full path to a span formated storage mounted on windows or directory containing the span files, the path should not end with a '\', payload size must be set, that it includes the zero termination

Payload Structure (read; write/out)

exported target id 32 Bits		
target index 1 Byte	**lun** 1 Byte	**reserved** 2 Byte
error 32 Bits		
path 1 1 Byte	N x 1 Byte	**path N** 1 Byte

exported target id

target id of the exported lun, specified by the directory path.

target index

target index of the exported lun, specified by the directory path.

lun

lun id of the exported lun, specified by the directory path.

error

Error code, in case of set path failed. Is always 0 in the payload of read direction. Only relevant for return payload of write direction.

Values:

no error	0
invalid directory path	1
invalid lun info file	2
common error	3

path 1 - N

path to a mounted span formatted storage or directory containing the span files.

CONF_SPAN_FILES

This command allows to set the path to a mounted span formatted storage or directory. You have to send the in payload to set the path. The min payload size is 16 bytes. The max path name length is 1024 bytes including the zero termination. payload size must be set to the size that includes the zero termination but min size is 16 bytes even in case of an empty string (at least zero termination). You can set up to 32768 pathes. Use the num param to specify the index of an entry. If you set the path it will check the path. To pass this check, at least a valid Lun info file (INFO.TXT) has to exist. The span are needed later when accessing the storage(e.g header access, replay). The response to the write direction is has the out payload format. If success the error field is zero and the payload includes the set path and the

exported lun address. This storage is now accessable by using this address for other commands. if the set failes, the error field is non zero and contains an error code. It is allowed to set path on an index that was already set. The old values will be overwritten, but remember that the effect takes place later, if there are still open references based on the old entry (e.g. running replay, header access).

The cmd can be used for read direction. The response payload contains the export lun address and the path. error code is always zero. If the entry is empty, the path is empty and has only the zero termination.

CONF_SPAN_HISTORY

Tag code	NumDes	Message	SNMP Support
0x0ace	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	read span history
Write	p_octet	access_right_service	clear span history or add a span history entry

Payload Structure

cam 16 Bits	**rec idx** 16 Bits
action 16 Bits	**cnt** 16 Bits
span address 1 8 Bytes ...	

N x 8 Bytes	**span address N** 8 Bytes ...

cam

camera line from 1 to max cam

rec idx

recording index from 1 to 2

action

action to do in this command, the command can show the span history, clear the complete history, add a new entry to the history, or show the remount span

Values:

show (read direction only)	0
add(write direction only)	1
clear(write direction only)	2
show remount(read direction only)	3

cnt

number of span address following in this command payload. For action show you can specifie the max number of span history entries to show for in payload. the reply payload will then return a max. of this number of span address. max 8.

span address 1 - N

range of span address from 0 to cnt, maximum 8.

span address content (8 bytes)

target id	4 bytes
target index	1 byte
lun	1 byte
span index	2 bytes

CONF_SPAN_HISTORY

This command is used by the device to store the span history in the device config, that means it adds (action add) an span adress entry to the config, for each recorded span. The history can be seen by read commando and action show. The history can be cleared by sending a action clear command.

CONF_ACCESS_LUN_MGMT_FILE

Tag code	NumDes	Message	SNMP Support
0x0ae2	yes	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	Access span header information.
Write	p_octet	access_right_service	Access lun management information.

Payload Structure

16 32

target id 32 Bits		
targ idx 8 Bits	**lun** 8 Bits	**reserved** 6 Bytes ...
data n Bytes(max 504 bytes)		

8 24

target id

target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

targ idx

target index of the lun

lun

lun

data

user data to read/write from/to the lun mgmt file

CONF_ACCESS_LUN_MGMT_FILE

This command can be used to acces the lun managment file. This exists only on span formatted luns, which were formatted by firmware /generic dll versions, that are able to create this file. Older versions cannot do this. The file name is "lunmgmt.bin". the max size which can be stored is 504 bytes. The in payload for reading or writing this command has to be at least a size of 12 bytes and includes the lun address for read/write from/to lun mgmt file. The response on the read direction includes the user data from the lun mgmt file. The data length will be the remayning bytes from the respons payload size minus the lun address and the reserved bytes (12 bytes together). Same in write direction. The payload size determines the number of bytes to write to the lun mgmt file(max 504 bytes).

CONF_FLUSH_LUN_INFO_CACHE

Tag code	NumDes	Message	SNMP Support
0x0b09	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported
Write	p_octet	access_right_service	flush the lun info cache of lun: 4 bytes IP as DWORD in network byte order, 1 byte target idx, 1 byte lun, 2 bytes reserved (if ip is 0.0.0.0 or 255.255.255.255 or the payload is less than 6 bytes it is asumed as wildcard, the whole cache will be flushed)

CONF_DELETE_CAM_REC_SPANS

Tag code	NumDes	Message	SNMP Support
0x0b5d	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported.
Write	p_octet	access_right_service	Delete recording spans of a camera from a span manager on a managed lun.

Request Payload Structure

16	32
Recording Camera 12 Bytes ...	
Lun Address 8 Bytes ...	
From 8 Bytes ...	
To 8 Bytes ...	
Flags 8 Bytes ...	

8 24

Recording Camera

The address of the camera for which recording spans are to be deleted.

16		32
IP 4 Bytes		
MAC 6 Bytes ...		
	Recording Index 1 Byte	**Camera** 1 Byte

8 24

IP

The ip address of the recording device.

MAC

The hardware address of the recording device. This field may be zero if you search for recordings without considering the mac.

Recording Index

1 = Primary Recording, 2 = Secondary Recording

Camera

The camera index.

Lun Address

The address of the lun on which recording spans for a camera will be deleted.

16		32
Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Reserved** 2 Byte

8 24

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES).

Target IDX

The index of the target.

LUN

The logical unit number.

From

Start time of the interval in Seconds since 2000 local time for deleting.

To

End time of the interval in Seconds since 2000 local time for deleting.

Flags

Values:

Bit 0	delete enclosed spans only
Bit 1	force delete

DELETE_CAM_REC_SPANS

This command is used for deleting spans of a specific recording line and index containing recordings between a given time interval. The flag "delete enclodes spans only" chooses wether a span shall be deletet if it is completetly enclosed by the time interval or if it only overlaps the interval. The flag "force delete" will delete all spans of the cam independend of the state of the span. That means even if the cam is still recording on a span, the span will be deleted. So use this option very carefully, as it could lead to inconsitencies. If the command return successfully, it doesn't mean, that the deletion is completetd, it just tells the caller, that a asychonous job for deletion was invoked. The command has to be send to the storage lun managing device.

CONF_HDD_VCD_CACHE_SIZE

	Tag code	NumDes	Message	SNMP Support
	0x0b79	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	see detailed description	
Write	p_octet	access_right_service	see detailed description,	

Payload Structure

16	32
line 16 bit	**rec idx** 16 bit
lower size limit 16 bit	**upper size limit** 16 bit
8	24

line

Line from 1 to n.

rec idx

Rec index: 1 - primary recording, 2 - secondary recording.

lower size limit

Lower size limit of used vcd cache buffer on recording spans in 64 kb units per 1 GB span. Allowed values from 1 to 4096, 0 to disable cache.

upper size limit

Upper size limit of used vcd cache buffer on recording spans in 64 kb units per 1 GB span. Allowed values from 1 to 4096, 0 to disable cache.

CONF_HDD_VCD_CACHE_SIZE

This command can be used to configure the size of the vcd cache on a span for recording. The size parameters are rellativ to 1 GB span size. If recording is running on soans with diffrent sizes than 1 GB, the values will be internally scaled to the propper size, so it will also work for diffrent span sizes. The usage of a vcd cache on spans will cause space overhead on a storage, so it is possible to switch it of by setting both values to 0. If enabled, the recording will use these values to adjust the size of the vcd cache on spans. The used adjusted sizes will be than limited by these two values, upper limit and lower limit. The adjusment starting value will be between these to values. While recording, the uses space for cached vcd data will be measured and after each span switch, the size will be adjusted by using the measurement values for new mounted spans in order to avoid overhead. It is also possible to set a fix size for the vcd cache by this command by setting the upper and lower limit to equal values. The default value for the command is 1 for lower limit and 128 for upper limit. For read direction an input payload is needed same as the described payload but only line and rec idx are required, the remaining part of the payload can be clipped away.

CONF_SPAN_CERTIFICATES_LIST

Tag code	NumDes	Message	SNMP Support
0x0c11	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_service	read certificates list from a span, see detailed description
Write	p_octet	access_right_service	not supported

Request

Payload Structure

16 32

Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Span IDX** 2 Byte
Start Time 4 Bytes		
Stop Time 4 Bytes		
TZ QH 1 Byte	**reserved** 3 Byte	
Max List Len 4 Bytes		

8 24

Target Id

Target ID of the Span

Target IdX

Target Index of the Span

LUN

LUN of the Span

Span IDX

Span Index of the Span

Start Time

Start time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

End Time

End time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Max List Len

Maximum size of the certificates list in response payload in bytes

Reply

Payload Structure

16 32

Target ID 4 Bytes		
Target IDX 1 Byte	**LUN** 1 Byte	**Span IDX** 2 Byte
Start Time 4 Bytes		
Stop Time 4 Bytes		
TZ QH 1 Byte	**Flags** 1 Byte	**reserved** 2 Byte
List Len 4 Bytes		
Certificates List max 16 kb		

8 24

Target Id

Target ID of the Span

Target IdX

Target Index of the Span

LUN

LUN of the Span

Span IDX

Span Index of the Span

Start Time

Start time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

End Time

End time of the intervall in seconds since 2000 local time based on the Timezone offset in TZ QH

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Flags

Values:

Bit 0	More certificates available, but the max length of the certificates list was exceeded

List Len

Size of the certificates list in response payload in bytes

Certificates List (sequence of)

16 32

Timestamp 4 Bytes		
TZ QH 1 Byte	**reserved** 1 Byte	**Length** 2 Byte
Certificate n bytes		

8 24

Timestamp

Local time in seconds since 2000

TZ QH

Timezone offset (from utc) in quarter hours as signed char value

Length

Length of the Certificates List Entry (certificate including these 8 bytes header infos)

Certificate

One certificate

CONF_SPAN_CERTIFICATES_LIST

This Command can be used to query the certificates of a span, in order to verify the signed video record data within that span. The response will deliver all certificates of the specified span and time intervall. The maximum size of the response can be limited by the caller but it will not exceed 16 kb for the certificates list. If the response payload size isnt't enough to hold all valid certificates, a flag in the response will signal the existence of further certificates, which didn't fit in the response payload. A second query in that case with a smaller time interval may help to get the remaining certificates. The response will have a certificates list with several enries of different size, there are no gaps between the entries.

CONF_STORAGE_REPORT

Tag code	NumDes	Message	SNMP Support
0x09cf	yes (1=cam1, 2=cam2, ...)	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get information about the storage device
Write	p_octet	access_right_service	not supported

Request packet

To get information about a storage device, a read packet with no payload must be send to the rcp server.

Response packet

The server reply will have the following structure:

Type 1 Byte	Availability 1 Byte	Access Type 1 Byte	IO Error Type 1 Byte
IO Error Count 4 Bytes			
Throughput Read 4 Bytes			
Throughput Write 4 Bytes			
iSCSI Login Connection 1 4 Bytes			

N x 4 Bytes	**iSCSI Login Connection N** 4 Bytes ...

Type

Values:

None	0x00
RAM Recording	0x03
iSCSI	0x04
USB	0x05
IDE	0x08
Compact Flash	0x09
SMB	0x0B

Availability

Values:

Unknown	0x00
No	0x01
Yes	0x02
locked by another device	0x03

Access Type

Values:

Unknown	0x00
Read Only	0x01
Read Write	0x02

IO Error Type

Values:

None	0x00
Read	0x01
Write	0x02

IO Error Count

Number of arised I/O errors.

Throughput Read

Kilobytes read from device.

Throughput Write

Kilobytes written to device.

iSCSI Login Connection 1 - N

Fields present only if device type is iSCSI

Sequence of:

16	32
Phase 2 Bytes	**Error** 2 Bytes
8	24

Phase

Indicates the phase, the iSCSI client has reached on login.

Error

Indicates the error that happend in that state.

CONF_STORAGE_REPORT_SECONDARY

	Tag code	NumDes	Message	SNMP Support
	0x0a50	yes (1=cam1, 2=cam2, ...)	yes	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	get information about the storage device of secondary recording, payload same as in command CONF_STORAGE_REPORT	
Write	p_octet	access_right_service	not supported	

CONF_STORAGE_MEDIUM_TYPE

	Tag code	NumDes	Message	SNMP Support
	0x09d3	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	(obsolete)	
Write	t_dword	access_right_service	set the storage medium type (None = 0, USB = 5, IDE = 8, CF = 9)(obsolete and not usable anymore)	

CONF_STORAGE_MEDIUM_AVAIL

	Tag code	NumDes	Message	SNMP Support
	0x09d4	no	no	no
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	list of dwords with available storage medium types (None = 0, USB = 5, IDE = 8, CF = 9, IMG File = 10, SD = 11, SPAN FILES = 12, SD NOT INSERTED = 14, CF NOT INSERTED = 15, IMG FILE NOT INSERTED = 16, SMB SPAN FILES = 17, SECOND SD = 18, SECOND SD NOT INSERTED = 19, SD OVER USB = 22, SD OVER USB NOT INSERTED = 23)	
Write	void	access_right_service	not supported	

CONF_STORAGE_LIST

	Tag code	NumDes	Message	SNMP Support
	0x0a37	no	yes	yes
	Datatype	**Access Level**	**Description**	
Read	p_octet	access_right_minimal	Read the device storage list.	
Write	p_octet	access_right_service	Write the stuff. (, not supported while recording is running)	

Payload Structure

Sequence of (max 16):

16 32

Storage Descriptor 20 Bytes ...

8 24

Storage Descriptor

Write Payload

16 32

Storage Type 4 Bytes			
Target ID 4 Bytes			
Target Index 1 Byte	**Lun** 1 Byte	**Rec Region Size** 2 Byte	
iSCSI export 1 Byte	**Span Manager** 1 Byte	**reserved** 2 Byte	
reserved 2 Bytes		**Flags** 1 Byte	**Storage threshold** 1 Byte

8 24

Read Payload

16 32

Storage Type 4 Bytes			
Target ID 4 Bytes			
Target Index 1 Byte	**Lun** 1 Byte	**Fmt Progress** 1 Byte	**Utilization** 1 Byte
iSCSI export 1 Byte	**Span Manager** 1 Byte	**Mgr Status** 1 Byte	**Mgr Error** 1 Byte
reserved 2 Bytes		**Flags** 1 Byte	**Storage threshold** 1 Byte

8 24

Storage Type

The type of the storage.

Values:

FILE	0x02
RAM	0x03
ISCSI	0x04
USB	0x05
IDE	0x08
CF	0x09
SD	0x0b
SMB	0x11
SD OVER USB	0x16

Target ID

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES). Will be ignored, if the storage type is not iscsi.

Target Index

The target index on the iscsi server. Will be ignored, if the storage type is not iscsi.

Lun

The iscsi lun. Will be ignored, if the storage type is not iscsi.

Rec Region Size

Size in GB from the Storage used for recording. Default is 0 which means as much recording space as the storage size allowes. The same will happen, if this Size is to big for the storage.

Fmt Progress

If the storage is in state formatting, the field indicates the formatting progress in percentage.

Utilization

If the storage is span formatted, this field indicates the amount capacity used (in percentage).

iSCSI export

Values:

For local use only	0x00
Make storage available through iSCSI	0x01

Span Manager

Values:

Span manager off	0x00
Span manager on	0x01
Span format lun and start span manager	0x02
Migrate lun and start span manager	0x03
read only mode	0x04
erase lun and format	0x05

Mgr Status

The status of the span manager in the response packet. Set to zero in the request packet.

Values:

OFF	0x00
ON	0x01
FORMATING	0x02
MIGRATING	0x03
ALARM THRESHOLD (msg only)	0x04
ALARM OVERWRITE (msg only)	0x05
OFFLINE (read payload only)	0x04
ONLINE (read payload only)	0x05
STOPPING	0x06
ERROR	0x07
ONLINE (WRITE PROTECTION)	0x08

Mgr Error

If the span manager status is ERROR, this field contains details of the error that occurred. See command CONF_SPAN_USE_STATUS for possible error codes.

Flags

Values:

default setting	0x01	just info, that this is a default setting

Storage threshold

This is the threshold of free storage in percent and means that at least this amount of memory should be free. A Message of this command will be send always, this treshold is crossed in eiher direction.

Use this command to configure and manage local or remote storage. The storage list may contain up to 16 entries. If a remote (iscsi) storage is to be managed, the lun address has to be specified. Realize, that if you export an remote storage, the device will have preformance drawbacks cause the ip- and iscsi-stack will be passed through twice. Prefer to access the remote storage directly instead.

The storage lun can be write protected. If this state was recognized, the Mgr Status will inform about this state by the value "WRITE PROTECTION". There are three ways for a storage lun to become write protected here. First to configure explicite the storage lun(Span Mgr) to be in read only mode, second by a defect storage lun, which will cause a fall back into a write protection mode, and third if the storage is set physically to write protection (e.g. user moves the lever on the sd card into the write protection position).

The storage devices will be exported through iscsi in the order of this list (storage devices with the 'iSCSI export' field set to 'For local use only' [0x00] are omitted) and the storage will appear in the same place of the target name list of an iscsi discovery on that device.

This command will be send as msg (can also be send as snmp trap with same pyload) in case of crossing the storage threshold with Mgr status ALARM THRESHOLD or in case of overwriting recording, that is still protected by unexpired retention time with mgr status ALARM OVERWRITE.

NOTE

If the storage list is empty or only contains entries with the 'iSCSI export' field set to 'For local use only' (0x00), the iscsi server is stopped, if it is running, cause no targets and luns would be provided. If the list contains at least one entry with the 'iSCSI export' field set to 'Make storage available through iSCSI' (0x01), the iscsi server is started, if not already running, to provide the iscsi service. The iscsi server can be controlled through the command CONF_ISCSI_SERVER_STATE too.

CONF_STORAGE_IO

Tag code	NumDes	Message	SNMP Support
0x0a61	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	List the currently running storage io tasks.
Write	p_octet	access_right_service	Start or stop a storage io task.

Request Packet Structure

16	32
Lun Address 8 Bytes ...	
Start Burst Size 4 Bytes	
End Burst Size 4 Bytes	
Burst Size Duration 4 Bytes	
Burst Size Increase 4 Bytes	
Repeat 4 Bytes	
Maximum Read Rate 4 Bytes	
Maximum Write Rate 4 Bytes	
Mode 2 Byte	**Flags** 2 Byte
ID 4 Bytes	
Action 1 Byte	**Reserved** 3 Bytes
Reserved 44 Bytes ...	
8	24

Lun Address

The address of the storage.

16		32
iSCSI IP 4 Bytes		
Target Idx 1 Byte	**Lun** 1 Byte	**Reserved or Span Idx** 2 Byte
8	24	

iSCSI IP

The ip address of the iscsi server

Target Idx

The index of the target on the iscsi server.

Lun

The logical unit.

Span Idx

The index of the span. Only used, if the flag RESTRICT TO SPAN is set.

Start Burst Size

The size in LBAs of the burst the test starts with. This value must be lower or equal to the end burst size.

End Burst Size

The size in LBAs of the burst the test ends with. This value may be clipped to the maximum burst value of the storage.

Burst Size Duration

The number of milliseconds each bust size is read and written.

Burst Size Increase

The number of LBAs the burst size is increased after each burst-duration.

Maximum Read Rate

The maximum datarate that is read in KBit/sec.

Maximum Write Rate

The maximum datarate that is written in KBit/sec.

Repeat

The number of times the test is repeated which means how often the specified burst sequence is run.

Mode

The test may run in the following modes. The modes RECORDING SIMULATION, LATENCY and REPLAY SIMULATION use different fields in the request packet which are not documented yet.

Values:

LINEAR	0x0000
RANDOM	0x0001
RECORDING SIMULATION	0x0002
LATENCY	0x0003
REPLAY SIMULATION	0x0004

If the mode is set to LINEAR then the test reads/write from/to consecutive lba addresses. If the maximum lba is reached and the STORAGE_IO_FLAG_MAX_LBA_END flag is set, the test stops. Otherwise the test wraps its lba pointer to the beginning of its storage.

If the mode is set to RANDOM then the test reads/write from/to randomly chosen lba addresses of the storage.

Flags

Flags to set some options.

Values:

READ ONLY	0x0001
WRITE ONLY	0x0002
KEEP DATA	0x0008
VERIFY WRITE	0x0010
EXP BURST INCREASE	0x0020
ERROR STOP	0x0040
RESTRICT TO SPAN	0x0080
ISCSI SESSION EXCLUSIVE	0x0100
MAX LBA END	0x0200

If the READ ONLY flag is set, only read operations are performed. May not be used simultaneously with the WRITE ONLY flag.

If the WRITE ONLY flag is set, only write operations are performed. May not be used simultaneously with the READ ONLY flag.

If the KEEP DATA flag is set, all lbas that are written are read before so that no recording data should be destroyed. Be careful that no one else writes to this lun simultaneously. In the latter case data on the storage could be lost.

If the VERIFY WRITE flag is set, every lba that was written is read afterwards to verify that the data was correctly written.

If the EXP BURST INCREASE flag is set, the field 'Burst Size Increase' is ignored. Instead the burst size is doubled after each rnu.

If the ERROR STOP flag is set, the tests stops when an io error occurrs.

If the RESTRICT TO SPAN flag is set, the tests only uses the span of the lun, which is specified in the 'Span Idx' field of the lun-address.

If the ISCSI SESSION EXCLUSIVE flag is set, the tests use an exclusive iscsi session which is not shared among other tests.

If the MAX LBA END flag is set, the tests stops when the write pointer reaches the maximum lba address of the lun/span (only in sequential access mode).

ID

Identifier of the storage io. Set to zero when starting a new storage io. Set to the from the start request returned value for stopping a storage io.

Action

Values:

STOP	0x00
START	0x01

Response and Message Packet Structure

16	32
Lun Address	

8 Bytes ...		
Start Burst Size 4 Bytes		
End Burst Size 4 Bytes		
Burst Size Duration 4 Bytes		
Burst Size Increase 4 Bytes		
Repeat 4 Bytes		
Maximum Read Rate 4 Bytes		
Maximum Write Rate 4 Bytes		
Mode 2 Byte		**Flags** 2 Byte
ID 4 Bytes		
Status 1 Byte	**Error** 1 Byte	**Reserved** 2 Bytes
Reapeat Count 4 Bytes		
Current Burst Size 4 Bytes		
Current Offset 4 Bytes		
Current Read Datarate 4 Bytes		
Current Write Datarate 4 Bytes		
Read Bytes 4 Bytes		
Write Bytes 4 Bytes		
Read Operations 4 Bytes		
Write Operations 4 Bytes		
Read Errors 4 Bytes		
Write Errors 4 Bytes		

8 24

Lun Address

See request packet structure for details.

Start Burst Size

See request packet structure for details.

End Burst Size

See request packet structure for details.

Burst Size Duration

See request packet structure for details.

Burst Size Increase

See request packet structure for details.

Repeat

See request packet structure for details.

Maximum Read Rate

See request packet structure for details.

Maximum Write Rate

See request packet structure for details.

Mode

See request packet structure for details.

Flags

See request packet structure for details.

ID

Identifier of the storage io. This value must be specified on actions other than start.

Status

Values:

STOPPING	0x00
STARTING	0x01
RUNNING	0x02
DONE	0x03
ERROR	0x04

Error

Values:

INTERNAL	0x01
LUN	0x02
INVALID PARAMS	0x03
READ	0x04
WRITE	0x05
WRITE VALIDATE	0x06

Repeat Count

The number of turns the test repeated so far.

Current Burst Size

The number of LBAs of the current burst.

Current Offset

The current position of the i/o pointer in percentage of the whole storage. This value is not significant in random access mode.

Current Read Datarate

The current read datarate in KBit/s.

Current Write Datarate

The current write datarate in KBit/s.

Read Bytes

The number of bytes that were read since the test was started.

Write Bytes

The number of bytes that were written since the test was started.

Read Operations

The number of read operations successfully performed since the test was started.

Write Operations

The number of write operations successfully performed since the test was started.

Read Errors

The number of read errors since the test was started.

Write Errors

The number of read errors since the test was started.

CONF_DATA_COPY_JOB_START

Tag code	NumDes	Message	SNMP Support
0x0b32	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	not supported ,
Write	p_octet	access_right_service	start copy job,

Payload Structure

16	32
source descriptor 16 to n Bytes ...	
...	
destination descriptor 16 to n Bytes ...	
...	
8	24

source descriptor

specifies the copy source (see src/dest descriptor payload)

destination descriptor

specifies the copy destination (see src/dest descriptor payload)

CONF_DATA_COPY_JOB_START

This command starts a copy job, given by first the source descriptor and second the destination descriptor. If starting the job was ok, the status will be reported by CONF_DATA_COPY_JOB_STATUS messages or can be querried by the same comand. The return payload structure is the same as the CONF_DATA_COPY_JOB_STATUS payload and contains the job id on success. This id can be used for stopping the job, query the state or to identify it in the status messages payload.

source/destination descriptor structure

16	32
offset 32 Bits	
len 32 Bits	
type 16 Bits	**type spec. params len (4 byte aligned)** 16 Bits
type specific parameter n Bytes	
...	
8	24

offset

the offset in lba sized units (= 512 bytes)

len

amount of data to copy in lba sized units (= 512 bytes), not relevant for destination descriptor

Values:

full copy	0
len	1 to n

type

type of the source or destination

Values:

interface test	0
span on iscsi	1
file	2
raw iscsi lun	3

type spec. params len

length of the following additional parameters

type specific parameter

additional parameter for specifing the source or destination depending on the type of the source or destination (see description for specific parameters below).

interface test

Fakes the access to a sorce/destination and can be used for example to test the interface and copy job engine. No further parameters are needed.

specific parameter structure for "span on iscsi"

16 32

target id 32 Bits		
target 8 Bits	**lun** 8 Bits	**span index** 16 Bits
file type 1 Byte	**reserved** 1 Byte	**timeout** 16 Bits

8 24

target id

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

target

Target index

lun

lun index

span index

span index

file type

type to specify, which part of the span has to be copied

Values:

lock header	1
unit header	2
manager header	3
recording data	4

timeout

request timeout in seconds, can be used to tollerate iscsi connection poblems

Values:

default, using default iscsi timeouts	0
possible values	1 - 65534
never timeout	65535

span on iscsi

Access to a span on an iscsi lun. This type can be used for example to copy a span. The parts of a span (unit, lock, mgr header, data) has to be copied in seperate copy jobs. Parameter length is 12 bytes.

specific parameter structure for "file"

16 32

file path n Bytes ...
...

8 24

file path

zero terminated ascii string specifing the path and file name of a source/destination

file

Access to a file. Can be used for example to copy a span to a file. If the file is the source, it has to exist. If the file is destination it will be created if it doesn't exist. Parameter length is the path length including zero termination and stuffing bytes in order to be 4 byte aligned. Size is at least 4 bytes.

specific parameter structure for "raw iscsi lun"

16 32

target id 32 Bits		

target	lun	timeout
8 Bits	8 Bits	16 Bits

8 24

target id

The target id of the lun (may be the ipv4 address for older versions or with default target resolve configuration, see CONF_TARGET_ID_RESOLVE_RULES)

target

Target index

lun

lun index

timeout

request timeout in seconds, can be used to tollerate iscsi connection poblems

Values:

default, using default iscsi timeouts	0
possible values	1 - 65534
never timeout	65535

raw iscsi lun

Access to a complete iscsi lun. This type can be used for example to make a one to one copy of a complete lun. Parameter length is 8 bytes.

CONF_DATA_COPY_JOB_STOP

Tag code	NumDes	Message	SNMP Support
0x0b33	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	not supported
Write	t_dword	access_right_service	requests stop on a data copy job if exists, otherwise fails, payload: job id

CONF_DATA_COPY_JOB_STATUS

Tag code	NumDes	Message	SNMP Support
0x0b34	no	yes	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the status of a copy job,
Write	p_octet	access_right_service	write not supported (only for read and messages),

Payload Structure

16 32

job id 32 Bits	
state 8 Bits	**reserved** 3 Bytes
progress 32 Bits	
full data amount 32 Bits	

8 24

job id

Id of a data copy job

state

state of the copy job

Values:

initializing	0
running	1
finished	2
failed	3
stopped	4

progress

Progress in lba units (= 512 bytes)

full data amount

Full data amount in lba units (= 512 bytes) of the copy job. Progress has to reach this value in order to finish successfully.

CONF_DATA_COPY_JOB_STATUS

This message reports the status of the copy job and will be send on each status change or several times while in state "running". The status can also be queried by sending this command in read direction.

CONF_STORAGE_TARGET_ID

Tag code	NumDes	Message	SNMP Support
0x0c09	no	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	get the storage target id, if there is no explicite configured (default 0), it returns the device unit ipv4 address
Write	t_dword	access_right_service	set the storage target id of this device which is an system wide unique identifier, which can be used to resolve it to connect to the local iscsi exported storage (0 means default)

CONF_TARGET_ID_RESOLVE_RULES

Tag code	NumDes	Message	SNMP Support
0x0c0a	no	no	no

	Datatype	Access Level	Description
Read	p_octet	access_right_minimal	get the target id resolve rules, see detailed description
Write	p_octet	access_right_service	set the target id resolve rules, see detailed description

Payload Structure

target id resolve rule 0 276 Bytes ...
...
target id resolve rule n 276 Bytes ...

target id resolve rule 1 - n

Upto 127 target id resolve rules. The payload contains all rules to be changed or to be added in write direction or all valid rules for read direction, at least one zeroed rule if there are no rules actual configured.

target id resolve rule structure

16 32

target id range start 4 Bytes		
target id range end 4 Bytes		
index 2 Bytes	**priority** 1 Bytes	**type** 1 Bytes
reserved 4 Bytes		
type specific parameter ... 256 Bytes		

...

8 24

target id range start

Begin of the target id range including the specified id of the rule.

target id range end

End of the target id range including the specified id of the rule.

index

Index of the target id resolve table, where to change the actual entry. Valid range is 0 to 127, as the table has an actual size of 128 entries.

priority

The rules target id range my overlap, therefor the priority will be used, which rule wins in conflic case. The valid range is from 0 to 255, where 0 is the lowest priority and 255 the highest.

priority

The rules target id range my overlap, therefor the priority will be used, which rule wins in conflic case. The valid range is from 0 to 255, where 0 is the lowest priority and 255 the highest.

Values:

NONE	0x00	no rule (delete rule)
DIRECT_IPV4	0x01	target id is the ipv4
ISCSI_SERVER	0x02	iscsi server parameter
SMB_SERVER	0x03	smb server parameter

ISCSI Server specific parameter

16 32

url ... 252 Bytes

...	
port 2 Byte	**reserved** 2 Byte

8 24

url

The url of the iscsi server as zero terminated ascii string. Only ipv4 as string is actual supported.

port

The port of the iscsi service socket on the server. Default 0 means use the default port. Actual it is just a place holder and other ports than the the default port are not supported.

SMB Server specific parameter

16 32

url ... 126 Bytes

...
path ... 126 Bytes

...	
port 2 Byte	**reserved** 2 Byte

8 24

url

The url of the smb server as zero terminated ascii string. Only ipv4 or ipv6 as string is actual supported.

path

The path on the smb server as zero terminated ascii string.

port

The port of the smb service socket on the server. Default 0 means use the default port. Actual it is just a place holder and other ports than the the default port are not supported.

CONF_TARGET_ID_RESOLVE_RULES

This command configures the rules for resolving a storage target id to the parameter for connecting the storage server. The target id is an 32 bit system wide unique identifier and replaces the ipv4 addresses for storages. The display format of an id in text will be "xxx.xxx.xxx.xxx" like ipv4 addresses. There are special values for the target id, which are 0.0.0.0 for invalid target id, 255.255.255.255 for wildcard and 127.0.0.1 for the local target in order to be backward compartible with older devices and software which expect an ipv4 address instead. The default state for the configured rules containes only one rule for the complete range 0.0.0.1 to 255.255.255.254 and lowest priority 0 for DIRECT_IPV4 for backard compartibility. In order to change or add rules, all which shall be changed or added needs to be send including the table entry index. Entries which shall not be changed doesn't need to be send. Also the read response payload will only contain valid rules, except if the table is empty, in that case only one empty rule will be in the payload.

NOTE

On the Generic (Windows and Linux) the whole table will be persisted in the table. The generic dll won't persist it. All other devices will only persist the first four valid rules.

CONF_INTERNAL_STORAGE_ENCRYPTION

Tag code	NumDes	Message	SNMP Support
0x0c43	no	no	no

	Datatype	Access Level	Description
Read	t_octet	access_right_minimal	enable : 1 or disable : 0 the xts encryption (no influence on device with permanent encryption)
Write	t_octet	access_right_service	enable : 1 or disable : 0 the xts encryption (no influenze on device with permanent encryption)

CONF_HDD_RECORD_ENCRYPTION

Tag code	NumDes	Message	SNMP Support
0x0c80	record index 1 (primary) or 2 (secondary)	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	hdd record encryption (span encryption only) : bit 0 -> 1ine 1; bit 1 -> line 2; .... bit n - 1 -> line n
Write	t_dword	access_right_service	enable the hdd encryption (span encryption only) : bit 0 -> 1ine 1; bit 1 -> line 2; .... bit n - 1 -> line n (0 - disabled (default streaming gateway), 1 - enabled (standard default)), additional a valid certificate for usage record encryption required in order to have a working record encryption

CONF_HDD_RECORD_ALARM_RING_INIT_SIZE

Tag code	NumDes	Message	SNMP Support
0x0cdf	-	no	no

	Datatype	Access Level	Description
Read	t_dword	access_right_minimal	initial alarm ring size on storage in MB (min 1 MB, max 128 MB, default 16 MB)
Write	t_dword	access_right_service	initial alarm ring size on storage in MB (min 1 MB, max 128 MB, default 16 MB)

CONF_HTTP_LIVE_BITRATE

	Tag code	NumDes	Message	SNMP Support
	0x0b45	no	no	no
	Datatype	**Access Level**	**Description**	
Read	t_dword	access_right_minimal	Get the bitrate in kBit the HTTP Live Streaming has allocated buffers for	
Write	t_dword	access_right_service	Set the bitrate in kBit the HTTP Live Streaming should allocate buffers for (max bitrate given in init gen)	

CONF_HTTP_LIVE_AUDIO

	Tag code	NumDes	Message	SNMP Support
	0x0b46	no	no	no
	Datatype	**Access Level**	**Description**	
Read	f_flag	access_right_minimal	Audio is enabled for HTTP Live Streaming yes or no	
Write	f_flag	access_right_service	Audio enabled for HTTP Live Streaming yes or no	

Appendix - Bicom Command Access Levels

The default protection level for BICOM / AUX commands is 'bl_user' for get-operations (get, getMax, getMin) and 'bl_service' for all other operations. All BICOM / AUX commands which have a protection level different from the default are listed in the following tables. 'bl_priv' means that the command is not accessible at all.

Device Server objects

ObjectID	Operation	Level
0x0160	BICOM_OP_GET	BL_SERVICE

Camera Server objects

ObjectID	Operation	Level
0x01B0	0x82 /*MoveCont*/	BL_USER
0x01B0	0x84 /*MoveMom*/	BL_USER
0x01F0	0x82 /*MoveCont*/	BL_USER
0x01F0	0x84 /*MoveMom*/	BL_USER
0x0430	BICOM_OP_SET	BL_PRIV
0x0430	BICOM_OP_SET_GET	BL_PRIV

PTZ Server objects

ObjectID	Operation	Level
0x0101	BICOM_OP_SET	BL_USER
0x0101	BICOM_OP_SET_GET	BL_USER
0x0103	0x81 /*Shot*/	BL_USER
0x0104	0x81 /*Shot*/	BL_USER
0x0110	BICOM_OP_SET	BL_USER
0x0110	0x82 /*MoveContFixSpeed*/	BL_USER
0x0110	0x83 /*MoveFixSpeedForPeriod*/	BL_USER
0x0110	0x85 /*MoveContVarSpeed*/	BL_USER
0x0110	0x88 /*MoveMomVarSpeed*/	BL_USER
0x0110	0x89 /*MoveContVarFinSpeed*/	BL_USER
0x0111	BICOM_OP_SET	BL_USER
0x0111	BICOM_OP_SET_GET	BL_USER
0x0112	BICOM_OP_SET	BL_USER
0x0112	BICOM_OP_SET_GET	BL_USER
0x0113	BICOM_OP_SET	BL_USER
0x0113	BICOM_OP_SET_GET	BL_USER
0x0114	BICOM_OP_SET	BL_USER
0x0114	BICOM_OP_SET_GET	BL_USER
0x0115	BICOM_OP_SET	BL_USER
0x0115	BICOM_OP_SET_GET	BL_USER
0x0116	BICOM_OP_SET	BL_USER
0x0116	BICOM_OP_SET_GET	BL_USER
0x011B	0x81 /*Shot*/	BL_USER
0x011C	0x81 /*Shot*/	BL_USER
0x011D	0x81 /*Shot*/	BL_USER
0x011E	0x81 /*Shot*/	BL_USER
0x0160	0x80 /*Record*/	BL_USER
0x0160	0x81 /*PlaybackCont*/	BL_USER
0x0160	0x82 /*PlaybackSingle*/	BL_USER
0x0160	0x83 /*Stop*/	BL_USER

0x0170	0x80 /*Record*/	BL_USER
0x0170	0x81 /*PlaybackCont*/	BL_USER
0x0170	0x82 /*PlaybackSingle*/	BL_USER
0x0170	0x83 /*Stop*/	BL_USER
0x01A1	BICOM_OP_SET	BL_USER
0x01A1	BICOM_OP_SET_GET	BL_USER
0x01A2	BICOM_OP_SET	BL_USER
0x01A2	BICOM_OP_SET_GET	BL_USER
0x01A2	0x0D /*GetHorizon*/	BL_USER
0x01A2	0x0E /*GetVerticalDown*/	BL_USER
0x01D0	BICOM_OP_SET	BL_USER
0x01D0	BICOM_OP_SET_GET	BL_USER
0x2000	BICOM_OP_SET	BL_USER
0x2000	BICOM_OP_SET_GET	BL_USER
0x2000	0x80 /*Store*/	BL_USER
0x2000	0x81 /*Shot*/	BL_USER
0x2000	0x83 /*GetPresetInfo*/	BL_USER
0x2000	0x84 /*Mask (new)*/	BL_USER
0x200x	BICOM_OP_SET	BL_USER
0x200x	BICOM_OP_SET_GET	BL_USER
.	BICOM_OP_SET	BL_USER
.	BICOM_OP_SET_GET	BL_USER
.	0x80 /*Store*/	BL_USER
.	0x81 /*Shot*/	BL_USER
.	0x83 /*GetPresetInfo*/	BL_USER
.	0x84 /*Mask (new)*/	BL_USER
0x2FF0	BICOM_OP_SET	BL_USER
0x2FF0	BICOM_OP_SET_GET	BL_USER
0x2FF0	0x80 /*Store*/	BL_USER
0x2FF0	0x81 /*Shot*/	BL_USER
0x2FF0	0x83 /*GetPresetInfo*/	BL_USER
0x2FF0	0x84 /*Mask (new)*/	BL_USER
0x2FFx	BICOM_OP_SET	BL_USER
0x2FFx	BICOM_OP_SET_GET	BL_USER

CA Server objects

ObjectID	Operation	Level

IO Server objects

ObjectID	Operation	Level
0x0400	BICOM_OP_EVENT_START	BL_PRIV
0x0400	0x71	BL_PRIV
0x0400	0x72	BL_PRIV
0x0400	0x73	BL_PRIV
0x0400	0x74	BL_PRIV
0x0400	0x75	BL_PRIV
0x0400	0x76	BL_PRIV
0x0400	0x77	BL_PRIV
0x0400	0x78	BL_PRIV
0x0400	0x79	BL_PRIV
0x0400	0x7a	BL_PRIV
0x0400	0x7b	BL_PRIV
0x0400	0x7c	BL_PRIV
0x0400	0x7d	BL_PRIV
0x0400	0x7e	BL_PRIV
0x0400	BICOM_OP_EVENT_END	BL_PRIV
0x0410	BICOM_OP_EVENT_START	BL_PRIV
0x0410	0x71	BL_PRIV
0x0410	0x72	BL_PRIV
0x0410	0x73	BL_PRIV
0x0410	0x74	BL_PRIV
0x0410	0x75	BL_PRIV
0x0410	0x76	BL_PRIV
0x0410	0x77	BL_PRIV
0x0410	0x78	BL_PRIV
0x0410	0x79	BL_PRIV
0x0410	0x7a	BL_PRIV
0x0410	0x7b	BL_PRIV
0x0410	0x7c	BL_PRIV
0x0410	0x7d	BL_PRIV
0x0410	0x7e	BL_PRIV
0x0410	BICOM_OP_EVENT_END	BL_PRIV
0x0800	BICOM_OP_SET	BL_USER
0x0800	BICOM_OP_SET_GET	BL_USER
0x0810	BICOM_OP_SET	BL_USER
0x0810	BICOM_OP_SET_GET	BL_USER
0x0820	BICOM_OP_SET	BL_USER
0x0820	BICOM_OP_SET_GET	BL_USER
0x0830	BICOM_OP_SET	BL_USER

0x0830	BICOM_OP_SET_GET	BL_USER

AUX numbers

Number	Level
1	BL_USER
2	BL_USER
7	BL_USER
8	BL_USER
50	BL_USER
51	BL_USER
52	BL_USER
53	BL_USER
65	BL_USER
78	BL_USER
81	BL_USER
82	BL_USER
83	BL_USER
84	BL_USER
100	BL_USER
101	BL_USER

SHOT numbers

Number	Level
1-99	BL_USER
101	BL_USER
102	BL_USER
201-456	BL_USER

SET numbers

Number	Level
1-99	BL_USER
201-456	BL_USER

Appendix - specific error codes

Each rcp command can generate an generic RCP fault as described in the chapter "Rcp Protocol Procedure". Additional command specific error codes are defined. If an command specific error occurs the Response is 2 bytes long. The first byte is set to RCP_ERROR_COMMAND_SPECIFIC (0xf0), the second byte is set to the specific fault.

The following command specific errors are defined:

Storage specific error codes

Span errors

SPAN_ERROR_INV_SPN_IDX	0x02
SPAN_ERROR_INV_SPN_IDX	0x02
SPAN_ERROR_INV_HDR_TYPE	0x03
SPAN_ERROR_INV_ADDR_LIST	0x04
SPAN_ERROR_NOT_MNTD	0x05
SPAN_ERROR_INV_FS	0x06
SPAN_ERROR_INV_LUN_NFO	0x07
SPAN_ERROR_BAD_HDR_CKSM	0x08
SPAN_ERROR_INV_IDX	0x09
SPAN_ERROR_RD_ONLY	0x0a
SPAN_ERROR_NO_REC_DAT	0x0b
SPAN_ERROR_INV_PART_NFO	0x0c
SPAN_ERROR_CONV_REC	0x0d
SPAN_ERROR_SPAN_REQUEST_FAILED	0x0e
SPAN_ERROR_SPAN_REQUEST_RETENTION_TIME	0x0f
SPAN_ERROR_REMOUNT_REFUSED	0x10
SPAN_ERROR_OLD_LUN_NFO	0x20

HDD PMM errors

HD_PMM_ERROR_TIMEOUT	0x12
HD_PMM_ERROR_CREATE_FAILED	0x22
HD_PMM_ERROR_ACCESS_DENIED	0x32
HD_PMM_ERROR_DEVICE_PRESENT_TIMEOUT	0x42
HD_PMM_ERROR_LUN_LOCK	0x52
HD_PMM_ERROR_INVALID_ACCESS	0x62
HD_PMM_ERROR_LUN_MGMT_FILE_NOT_FOUND	0x72
HD_PMM_ERROR_LUN_WRITE_PROTECTED	0x82
HD_PMM_ERROR_COMMON	0xf2

ISCSI errors

ISCSI_ERR_CONNECT	0x31
ISCSI_ERR_INV_LUN	0x33
ISCSI_ERR_LOGIN	0x34
ISCSI_ERR_INV_TARG_IDX	0x35
ISCSI_ERR_PWD	0x36
ISCSI_ERR_PROTO	0x37
ISCSI_ERR_TARG_NOT_REACH	0x38
ISCSI_ERR_NO_MEMORY	0x3a
ISCSI_ERR_SESS_CREATE	0x3b
ISCSI_ERR_INV_PARAMS	0x3c
ISCSI_ERR_SESS_NOT_FOUND	0x3d
ISCSI_ERR_DISCONN	0x3e
ISCSI_ERR_TIMEOUT	0x3f
ISCSI_ERR_TARGET_NOT_SUPP	0x40
ISCSI_ERR_TARGET_SESSION_LIMIT	0x41
ISCSI_ERR_CMD_NOT_SUPP	0x42
ISCSI_ERR_TARGET_NOT_FOUND	0x43
ISCSI_ERR_SOCK	0x5f
ISCSI_ERR_TARG_PM	0x6f
ISCSI_SOCK_CLOSED	0x7f
ISCSI_ERR_TCP_CONN_RESET	0x8f
ISCSI_ERR_INTR_NOT_SUPP	0x9f
ISCSI_ERR_IP_ZERO	0xa0
ISCSI_ERR_OUT_OF_RESOURCES	0xa1