VIDOS Automation Interface

BOSCH

en Reference Manual

Table of Contents

1 Introduction

The VIDOS Automation Interface provides means for remote control and supervision via a socket interface. Two basic mechanisms are provided, one for issuing synchronous commands and one for receiving event information.
The Automation Interface can be accessed via TCP port 1758. It supports a large number of clients in parallel for sending commands but only a single listener for events.

2 Authentication

The first line sent to the client shall contain the keyword ‘login’ followed by the user name ‘script’ and the password of the user script.
In order to avoid any security risks scripting is disabled as long as no user named ‘script’ has been defined. When the user is defined without any password the login procedure is optional. With a non-empty password, the password provided in the login command must match, otherwise the connection terminates.

3 Command Interface

3.1 General Syntax

VIDOS responds to each received command line with one response line. Lines are terminated by a carriage return and a new line (’\r\n’).
Commands are passed as ASCII strings as C- or Perl-style functions. Strings must be enclosed by double quotes. Integer values are passed without any markup.

3.1.1 Object Identifier

Elements in VIDOS Automation Interface can be addressed in several ways.

- The device IP address followed by a type identifier and the enumerator. Valid type identifiers are ’camera’, ‘monitor’, ’digital-out’ and ‘digital-in’.
  Example for addressing the third input of a device: "10.0.12.25/camera/3".
- The name of the object. This applies to camera inputs, monitor outputs, digital inputs, digital outputs, manual triggers, software monitors and site maps.
  Example for addressing the entrance of the fourth monitor: "Entrance Monitor 4".
- A number sign followed by the globally enumerated number as it is used for CCTV keyboard operation.
  Example for addressing the 23rd camera: "#23".

3.1.2 Responses

On successful operation the commands return a single line containing either an integer or a string.
In case of any error, the returning line starts with the keyword 'Error' followed by an error code and detailed textual information.
The following error codes are defined:

Error Code	Description
1	Unknown object
2	Unknown function
3	Wrong number of arguments
4	Invalid argument
5	Invalid command syntax
6	Unexpected error during execution

3.2 Video and Audio Switching

3.2.1 connect

Definition

int connect(source, destination);

Parameters

`source` – An object identifier for a camera or salvo
`destination` – An object identifier for a hardware or software monitor

Description

Sets up a streaming connection between the specified source and the destination. The call immediately returns after having verified the object identifiers. If monitoring of the connection is required, this has to be implemented via the event interface.

Return

‘0’ on success or any of the defined errors

3.2.2 disconnect

Definition

int disconnect(destination);

Parameters

`destination` – An object identifier for a hardware or software monitor

Description

Terminates a streaming connection to the specified destination.

Return

‘0’ on success

3.2.3 getCamera

Definition
string getCamera(monitor);

Parameters
`monitor` – An object identifier for a software monitor

Description
Gets the identifier for the camera currently displayed on the specified monitor. The method call returns a no object error if the monitor is unconnected.

Return
Identifier for the attached camera in standardized notation
'NoObjectError' – Either the monitor identifier does not correspond to an existing monitor or no camera is connected to the monitor

3.2.4 setAudio

Definition
int setAudio(monitor, mode);

Parameters
`monitor` – An object identifier for a software monitor
`mode` – See description below

Description
Enables or disables audio reception and sending for the specified monitor. The following modes are available:
"0" – No audio transmission
"1" – Decode and output audio to the local speakers
"2" – Send audio from the local microphone to the remote site
"3" – Full duplex audio enabled

Return
'0' on success

3.2.5 setEncoderPreset

Definition

int setEncoderPreset(camera, preset);

Parameters

`camera` – An object identifier for a camera
`preset` – The number of the preset

Description

Modifies the preset of an encoder.
Typically the preset value may be in the range of ‘1’ to ‘8’.

Return

‘0’ on success

3.2.6 saveSnapshot

Definition

int saveSnapshot(monitor);

Parameters

`monitor` – An object identifier for a software monitor

Description

Saves a snapshot in the local snapshot database.

Return

‘0’ on success

3.2.7 changeMonitorLayout

Definition

int changeMonitorLayout(monitor, array, thumbnails=0);

Parameters

`monitor` – An object identifier of a hardware monitor
`array` – The number of monitors in a row/column
`thumbnails` – The number of (smaller) monitors surrounding the array

Description

Changes the monitor layout of a hardware monitor.
The parameters ‘array’ and ‘thumbnails’ define the new layout of the monitor. Currently only VIDOS Monitor wall supports the thumbnail option.
Examples:
changeMonitorLayout("#1", 2);

– changes a monitor to a 2x2 array

changeMonitorLayout("#1", 1, 5);

– changes a hardware monitor to a layout with one big screen and 5 surrounding thumbnails

Return

‘0’ on success
This does not mean that the layout actually changed. It only means that the command has been sent to the device.

3.3 Controlling Peripheral Devices

3.3.1 movePtz

Definition

int movePtz(monitor, pan, tilt, zoom);

Parameters

`monitor` – An object identifier for a hardware or software monitor

`pan` – Pan speed in 1/100 of degree per second (range is +/- 8000)

`tilt` – Tilt speed in 1/100 of degree per second (range is +/- 5000)

`zoom` – Zoom speed in percent

Description

Moves the dome attached to the specified monitor.
Don't forget to stop the motion by issuing the command with all parameters set to zero.
The method also supports direct addressing of cameras. However this mode is only available for those IDs that do not overlap with any existing monitor IDs.

Return

'0' on success
'-1' on error, if for example no controller is specified for this camera

3.3.2 execPeripheral

Definition

int execPeripheral(monitor, command, value);

Parameters

`monitor` – An object identifier for a hardware or software monitor

`command` – Action to be executed (see description below)

`value` – Index or speed value (see description below)

Description

Executes a command on a peripheral device.

The command identifier must be passed as a string value.

The following commands are supported:

`focus` – Change the lens focus, the value is interpreted as speed in the range of +/- 100%

`iris` – Change the lens iris, the value is interpreted as speed in the range of +/- 100%

`gotoPreset` – Restore a camera internal preset position, the value identifies the preset number

`storePreset` – Save a camera internal preset position, the value identifies the preset number

`aux` – Execute an auxiliary function, the value identifies the function number

`auxOn` – Execute an auxiliary function (for Bosch AutoDome only)

`auxOff` – Execute an auxiliary function (for Bosch AutoDome only)

The method also supports direct addressing of cameras. However this mode is only available for those IDs that do not overlap with any existing monitor IDs.

Index values start from ‘1’, ‘0’ or negative values are not allowed.

Return

‘0’ on success

‘-1’ on error, if for example no controller is specified for this camera

3.4 Control of Digital IOs and Alarms

3.4.1 set

Definition
int set(output, value);

Parameters
`output` – An object identifier for a digital output or a manual trigger
`value` – Either ‘0’ or ‘1’

Description
Sets or resets a digital output relay or a VIDOS internal manual trigger.

Return
‘0’ on success

3.4.2 get

Definition
int get(input);

Parameters
`input` – An object identifier for a digital input, digital output or a manual trigger

Description
Reads the current setting of a digital input.
This method may also be applied to digital outputs and manual triggers.

Return
IO state which is either ‘0’ or ‘1’

3.5 Control of the Site Map and its Controls

3.5.1 changeSitemap

Definition
int changeSitemap(sitemap);

Parameters
`sitemap` – String containing the name or number of a site map

Description
Displays the specified site map replacing any previously displayed site map. Together with the two special parameters this command allows a browser like site map navigation.
Two special parameters are provided:
`"#back"` – display the previous site map
`"#forward"` – if available display the next site map

Return
‘0’ on success

3.5.2 getVisibleControls

Definition
string getVisibleControls();

Parameters
<none>

Description
Retrieves the names of all interactive objects on the currently visible site map. The returned hash table is formatted in Perl-style. The number corresponds to the ‘kbid’ property and the name to the ‘name’ property.
This method allows the construction of context dependent user interfaces that can be activated using the ‘activateControl’ method.
Only those elements of the visible site map are shown that have a defined action, kbid and name property.

Return

A hash list with pairs of keyboard ID and name
Example for the return string:
‘{ "1" => "On", "2" => "Off", "44" => "Back"}’

3.5.3 activateControl

Definition

int activateControl(element, mode);

Parameters

`element` – String containing the name or number of the element
`mode` – One of ‘0’, ‘1’ or ‘2’ (default)

Description

Executes a user interface element action. A source may either press (‘1’), depress (‘0’) or pulse (default) the element.
Only visible user interface elements can be activated.

Return

‘0’ on success

Note:

The initial implementation does not support the optional mode parameter.

3.6 Accessing the Alarm Stack

3.6.1 processAlarm

Definition

int processAlarm(command);

Parameters

`command` – Action to be executed (see description below)

Description

Allows changing the state of the alarm at the top of the alarm stack.

Valid commands are:

`activate` – Activate the alarm (depending on the programmed behaviour this may trigger alarm jobs)

`confirm` – Acknowledge and clear an activated alarm

`toggle` – Advance the alarm state from pending to activated or from activated to cleared

Return

‘0’ on success

3.7 Instant Replay

3.7.1 replay

Definition

int replay(monitor, command, value);

Parameters

`monitor` – Software monitor to be controlled
`command` – Action to be executed (see description below)
`value` – Optional numerical value (see description below)

Description

This command allows controlling replay of cameras that are available as live view. Before issuing any other commands the monitor must be set to replay mode by issuing the 'enable' command. Then replay speed can be changed using the 'play' command with variable speed parameter. In pause mode the display can be stepped using the 'seekFrame' command.
Supported commands:
`enable` – Change the monitor display to 'Instant Replay'
`disable` – Switch the monitor back to live display
`play` – Change the replay pace to the value given in percent of real-time, a value of '0' corresponds to pause
`seekFrame` – Seek the next frame (value=1) or the previous I-frame (value=-1)

Return

'0' on success

3.8 System Commands

3.8.1 logoff

Definition

int logoff(reason = "");

Parameters

`reason` – A string that will be displayed in the login dialog

Description

Closes the current session and displays the login dialog.

Return

‘0’ on success

4 Event Subscription and Deliver

VIDOS provides a subscription interface. Sending a command line starting with the keyword ‘subscribe’ followed by one or more event types subscribes for the given events. These will be delivered asynchronously until the subscription is changed via a subscribe command with different event types. Subscription can be disabled by providing an empty type list

Event Type	Description
error	Error conditions
info	Basic information
user	User logon and logoff
alarm	Alarm task processing
trigger	Trigger state changes
connection	Video connection changes

The information is provided in exactly the same format as it is written to the log file.

Example:

[2009-06-16T17:50:31] [trigger] [on] [admin] "160.10.12.104/camera/1/motionAlarm" "Name of Motion Alarm"

The following trigger identifiers are supported:

- Digital Input: <IP-Address>/digital-in/<no>
- Digital Output: <IP-Address>/digital-out/<no>
- Motion Alarm: <IP-Address>/camera/<line>/motionAlarm
- Video Loss: <IP-Address>/camera/<line>/videoLoss
- VCA Alarm #1: <IP-Address>/camera/<line>/vcaAlarm/1
- VCA Alarm #2: <IP-Address>/camera/<line>/vcaAlarm/2
- VCA Alarm #3...16: <IP-Address>/camera/<line>/vcaAlarm/3-16
- Tamper Alarm: <IP-Address>/camera/<line>/tamperAlarm

The following connection identifiers are supported:

- Camera Input: <IP-Address>/camera/<line>/<encoder>
- Monitor Output: <IP-Address>/monitor/<line>
- VIDOS Monitor: vidos/monitor
- VIDOS Monitor with assigned keyboard Id: vidos/monitor/<kbId>

Event dispatching is restricted to a single client. The client must continuously read the socket, otherwise event messages may get lost.

5 Configuring Keyboard IDs in VIDOS

Addressing via the so called keyboard IDs is mandatory for using VIDOS with CCTV keyboards, but may also be used throughout the scripting interface. During system configuration unique IDs must be assigned to the entities of each type. The keyboard IDs can be assigned and modified via properties. The following list shows which properties are available and where they are to be found:

Type	Location
Camera	‘Properties’ sheet in the ‘Resources’ tree
Control Action	Property ’kbid’ in the ‘Site Map’ editor
Digital Output	‘Properties’ sheet in the ‘Resources’ tree
Hardware Monitor	‘Properties’ sheet in the ‘Resources’ tree
Manual Trigger	‘Properties’ sheet in the ‘Resources’ tree
Salvo	‘Properties’ sheet located in the ‘Configuration’ window
Site Map	‘Properties’ sheet located in the ‘Configuration’ window
Software Monitor	Property ‘kbid’ in the ’Site Map’ editor

Glossary

H

Hardware Monitor
: External monitor typically connected to a VIP XD or a VIP X1600 XFMD.

S

Salvo
: Sequencer that switches video inputs periodically.

Software Monitor
: Monitor located in a sitemap.

System NVR
: NVR specified via the InstantReplay settings.

Bosch Sicherheitssysteme GmbH
Robert-Koch-Straße 100
D-85521 Ottobrunn
Germany
Telefon 089 6290-0
Fax 089 6290-1020
www.boschsecurity.com