

VAPIX® version 3

Event Handling

Copyright Notice

This document is copyright protected and is the property of Axis Communications AB and may not be copied, reproduced or distributed in any way without the prior written consent of Axis Communications AB.

VAPIX® LICENSE AGREEMENT

This VAPIX® License Agreement ("License") is a legal agreement between you (either individual or an entity) and Axis Communications AB ("Axis"). By using the INTERFACE and INTERFACE DESCRIPTION (each defined below), whether in whole or in part, you agree to be bound by the terms of this License.

1. GRANT OF LICENSE

Axis hereby grants to you the right to use the AXIS VAPIX application programming interface ("INTERFACE") and the written specification of the INTERFACE (the "INTERFACE DESCRIPTION") for the sole and limited purpose of creating, manufacturing and developing a solution that integrates any unit or portion included in the product range of Axis network products, as defined by Axis at its discretion (an "Axis Product") and to market, sell and distribute any such solution.

2. COPYRIGHT

The INTERFACE and the INTERFACE DESCRIPTION are owned by Axis and are protected by copyright laws and international treaty provisions. Any use of the INTERFACE and/or INTERFACE DESCRIPTION outside the limited purpose set forth in Section 1 above is strictly prohibited.

3. RESTRICTIONS ON USE

You have no rights with respect to the INTERFACE, INTERFACE DESCRIPTION or any portions thereof and shall not use the INTERFACE, INTERFACE DESCRIPTION or any portion thereof except as expressly set forth herein. You may not reverse engineer, decompile, or disassemble the INTERFACE except to the extent required to obtain interoperability with other independently created computer programs as permitted by mandatory law.

4. THIRD PARTY RIGHTS

You agree that you are fully responsible for your own conduct while using the INTERFACE and integrating any Axis Products into your solution and the consequences thereof. Axis Products may be combined with a virtually infinite number of potential solutions. Consequently, you recognize that (i) other third parties may claim to own patents or copyrights that could cover certain solutions which integrate Axis products, or which result from the combination of Axis products and additional technology or solutions and (ii) you are responsible for ensuring that any solution which integrates with an Axis Product, or a combination of a solution and an Axis product, does not infringe upon or misappropriate any intellectual property or personal right of any third party.

5. TERMINATION

This License is effective until terminated. Your rights under this License will terminate automatically without notice from Axis if you fail to comply with any term(s) of this License. Upon the termination of this License, you shall cease all use and disposition of the INTERFACE and/or INTERFACE DESCRIPTION whether for the purpose set forth in Section 1 above or not.

6. REPRESENTATIONS AND WARRANTIES; DISCLAIMER

- 6.1. You represent and warrant that (i) any solution created, manufactured and/or developed by you which integrates an Axis Product shall not infringe or otherwise violate any third party rights, including but not limited to third party intellectual property rights; and (ii) your use of the INTERFACE and INTERFACE DESCRIPTION will comply with all applicable foreign and domestic laws, rules and regulations.
- 6.2. YOUR USE OF THE INTERFACE IS AT YOUR SOLE RISK. THE INTERFACE AND THE INTERFACE DESCRIPTION ARE DELIVERED FREE OF CHARGE AND "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK AS TO THE USE, RESULTS AND PERFORMANCE OF THE INTERFACE AND THE INTERFACE DESCRIPTION IS ASSUMED BY THE USER/YOU. AXIS DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT AND PRODUCT LIABILITY, OR ANY WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE WITH RESPECT TO THE INTERFACE AND THE INTERFACE DESCRIPTION. Without limiting the generality of the foregoing, you acknowledge and agree that Axis does not make any representation or warranty that the integration of Axis Products into your solution does not infringe any third party rights. You are solely responsible for any intellectual property infringement claims that are based on or relate to solutions created, manufactured and distributed by you which integrate Axis Products. Axis is unaware of the details regarding your particular solution, has not conducted any investigation relating to potential third party rights issues relating to your solution and does not accept any responsibility or liability with respect thereto.
- 6.3. THIS LICENSE DOES NOT CONVEY ANY LICENSE TO THIRD PARTY INTELLECTUAL PROPERTY. YOU ARE SOLELY RESPONSIBLE FOR (I) EXAMINING WHETHER THE INTERFACE AND THE INTERFACE DESCRIPTION ARE ENCUMBERED BY OR INFRINGES UPON A RIGHT HELD BY A THIRD PARTY AND (II) ANY INTELLECTUAL PROPERTY INFRINGEMENT CLAIMS THAT ARISE OUT OF OR RELATE TO SOLUTIONS CREATED, MANUFACTURED AND DISTRIBUTED BY YOU WHICH INTEGRATE AXIS PRODUCTS.

7. LIMITATION OF LIABILITY

- 7.1. AXIS SHALL NOT BE LIABLE FOR LOSS OF DATA, LOSS OF PRODUCTION, LOSS OF PROFIT, LOSS OF USE, LOSS OF CONTRACTS OR FOR ANY OTHER CONSEQUENTIAL, ECONOMIC OR INDIRECT LOSS WHATSOEVER IN RESPECT OF USE OR DISPOSITION OF THE INTERFACE AND THE INTERFACE DESCRIPTION.
- 7.2. AXIS TOTAL LIABILITY FOR ALL CLAIMS IN ACCORDANCE WITH THE USE OF THE INTERFACE AND THE INTERFACE DESCRIPTION SHALL NOT EXCEED THE PRICE PAID FOR THE INTERFACE AND THE INTERFACE DESCRIPTION.
- 7.3. YOU UNDERTAKE NOT TO PURSUE ANY CLAIMS WHATSOEVER AGAINST AXIS OR ITS AFFILIATES RELATING TO OR EMANATING FROM THE INTERFACE AND THE INTERFACE DESCRIPTION OR YOUR INTEGRATION OF AN AXIS PRODUCT INTO YOUR SOLUTION.

8. INDEMNIFICATION

You will indemnify and hold Axis, its subsidiaries, affiliates, officers, employees, and agents harmless from any and all claims, damages, losses, liabilities, actions, judgments, costs, and expenses brought by a third party, including claims for infringement of intellectual property rights, arising out of or in connection with (i) your use of the INTERFACE or INTERFACE DESCRIPTION other than in accordance with the terms of this agreement, and/or (ii) any solution created, manufactured and/or developed by you which integrates an Axis Product.

9. GOVERNING LAW

This agreement shall be deemed performed in and shall be construed by the laws of Sweden. All disputes in connection with this agreement shall be finally settled by arbitration in accordance with the Rules of the Arbitration Institute of the Stockholm Chamber of Commerce. The place of arbitration shall be Malmö, Sweden. The language of the proceedings, documentation and the award shall be English.

Table of Contents

- VAPIX® LICENSE AGREEMENT 2
- 1 Overview 7
 - 1.1 Description 7
 - 1.2 History 7
- 2 Prerequisites 7
 - 2.1 Identification 7
- 3 Common Examples 8
 - 3.1 Create events 8
 - 3.2 Event actions 9
 - 3.3 Event servers 10
 - 3.4 A complete event handling example 10
- 4 Event Handling and the HTTP API 12
 - 4.1 Events 12
 - 4.2 Event actions 13
 - 4.2.1 Dynamic event notification messages 13
 - 4.3 Event servers 14
- 5 Parameters 15
 - 5.1 Events 15
 - 5.1.1 Software input triggers 18
 - 5.2 Event actions 20
 - 5.2.1 FTP Actions 20
 - 5.2.2 HTTP Actions 21
 - 5.2.3 HW Actions 23
 - 5.2.4 SMTP Actions 24
 - 5.2.5 TCP Actions 25
 - 5.2.6 PTZ Actions 26
 - 5.2.7 Guard tour actions 27
 - 5.2.8 Auto tracking actions 27
 - 5.2.9 Light control actions 28

- 5.3 Event servers 29
 - 5.3.1 FTP Servers 29
 - 5.3.2 HTTP Servers 30
 - 5.3.3 TCP Servers 31
- 5.4 SMTP Servers 31
 - 5.4.1 SMTP Authentication 31
- 6 References 32

©2009-2013 Axis Communications AB. AXIS COMMUNICATIONS, AXIS, ETRAX, ARTPEC and VAPIX are registered trademarks or trademark applications of Axis AB in various jurisdictions. All other company names and products are trademarks or registered trademarks of their respective companies. We reserve the right to introduce modifications without notice.

1 Overview

1.1 Description

This document describes the event handling feature in Axis network cameras and video encoders with firmware 5.00 and later. An event is a set of rules describing how and when the network video product should perform certain actions. As an example, a network camera can be configured to upload images to an FTP server and send a notification message when motion is detected in the image.

Notes:

- Support for event triggers and actions vary between product models.
- For improved performance, the event script has been removed.

1.2 History

Version	Date	Comment
1.00	2009-May-13	Initial version
1.10	2013-July-09	New VAPIX® License Agreement

2 Prerequisites

2.1 Identification

Property: Properties.API.HTTP.Version=3

Firmware: 5.00 and later

See the product's Release notes for supported dynamic parameter groups and available templates.

3 Common Examples

3.1 Create events

Example 1: Create a new event. Events are dynamic parameter groups; the template and the parent group must be specified in the request.

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event&template=event
```

Response: The character E comes from the first character in the parent group (i.e. Event). The group number, 0, is dynamically added by the product. The new event group is Event.E0. Note that you must parse the response to know which number the event group got.

```
E0 OK
```

Example 2: List the parameters in the Event.E0 group.

```
http://myserver/axis-cgi/param.cgi?action=list
&group=Event.E0
```

Response: No parameter values have been specified yet, so all parameters have their default values. (Only a subset of the event parameters is shown below.)

```
root.Event.E0.Name=New Event
root.Event.E0.Type=T
root.Event.E0.Enabled=yes
root.Event.E0.Priority=1
root.Event.E0.Image=0
root.Event.E0.HWInputs=xxxx
root.Event.E0.SWInput=
root.Event.E0.Weekdays=1111111
root.Event.E0.Starttime=00:00
root.Event.E0.Duration=24:00
...
```

Example 3: Configure the event Event.E0 to trigger when motion is detected in motion window 0 (%20 is the URL code for space).

```
http://myserver/axis-cgi/param.cgi?action=update
&Event.E0.Name=Motion%20Detection
&Event.E0.SWInput=M0: /
```

Response:

```
OK
```

Example 4: Event parameters can be modified at the same time as the event is created. The group number is not known at this time and omitted from the request. This event is activated when the analog video signal is lost on channel 1 (applies to video encoders only).

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event&template=event
&Event.E.Name=VideoLossEvent
&Event.E.SWInput=V0:\
```


Response:

```
E1 OK
```

Example 5: Create a scheduled event that starts at 17:00 (5 pm) every Friday and continues until Monday 08:00 (8 am).

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event&template=event
&Event.E.Name=Weekends
&Event.E.Type=S
&Event.E.Weekdays=0000010
&Event.E.Starttime=17:00
&Event.E.Duration=63:00
```

Response:

```
E2 OK
```

Tip: The Event parameter group contains a lot of parameters. Use wildcards to list, for example, only the event names:

```
http://myserver/axis-cgi/param.cgi?action=list
&group=Event.*.Name
```

Response example:

```
root.Event.E0.Name=Motion Detection
root.Event.E1.Name=VideoLossEvent
root.Event.E2.Name=Weekends
root.Event.E3.Name=DigitalInputEvent
```

3.2 Event actions

Example 1: Add an action of type FTP upload to the event Event.E0.

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event.E0.Actions&template=ftpaction
```

Response: The new action is Event.E0.Actions.A0. The character A comes from the first character in the subgroup Actions.

```
A0 OK
```

Example 2: Connect the group Event.E0.Actions.A0 to the event servers F0 and F1 (i.e. EventServers.FTP.F0 and EventServers.FTP.F1)

```
http://myserver/axis-cgi/param.cgi?action=update
&Event.E0.Actions.A0.Server=F0
&Event.E0.Actions.A0.Server2=F1
```

Example 3: Add a TCP action to the event Event.E0. Use a pre-configured TCP event server and send a dynamic event notification message using the event description modifier #E (%23 is the URL-code for the # character). The modifier will be replaced with "Motion '%s' on motion window '%s' " (here the first %s will be replaced with "started" or "stopped" and the second %s with motion window name).

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event.E0.Actions&template=tcpxaction
&Event.E0.Actions.A.Server=T0
&Event.E0.Actions.A.Message=%23E
```

Response: This is the second action, Event.E0.Actions.A1, of Event.E0.

```
A1 OK
```

3.3 Event servers

Example 1: Create a TCP event server, set the IP address and port number in the same request.

```
http://myserver/axis-cgi/param.cgi?action=add
&group=EventServers.TCP&template=tcp_config
&EventServers.TCP.T.Address=10.13.24.10
&EventServers.TCP.T.Port=4444
```

Response

```
T0 OK
```

Example 2: Configure e-mail settings. Note that the SMTP server is not an event server.

```
http://myserver/axis-cgi/param.cgi?action=update
&SMTP.FromEmail=MyCamera@example.com
&SMTP.MailServer1=10.13.0.5
&SMTP.MailServer2=10.13.0.6
```

3.4 A complete event handling example

In this example, we will configure the network video product to upload images to an HTTP server when digital input 1 is activated.

Step 1: Add an HTTP server, set the IP address and port number in the same request.

```
http://myserver/axis-cgi/param.cgi?action=add
&group=EventServers.HTTP&template=http_config
&EventServers.HTTP.H.Name=MyHTTPServer
&EventServers.HTTP.H.Address=http://192.168.254.10/cgi-bin/upload.cgi
&EventServers.HTTP.H.Login=user
&EventServers.HTTP.H.Password=pass
```

Response: The new group is EventServers.HTTP.H0.

```
H0 OK
```

Step 2: Create an event that is triggered when digital input 1 is activated. Let the event be active Monday through Friday, include 3-seconds pre-trigger and post-trigger buffers and use the image settings saved in the stream profile "MyProfile". The buffers and image settings are used for image upload (see next step).

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event&template=event
&Event.E.Name=IOEvent
&Event.E.HWInputs=1x
&Event.E.Weekdays=0111110
&Event.E.IncludePreTrigger=yes
&Event.E.PreTriggerDuration=3000
&Event.E.IncludePostTrigger=yes
&Event.E.PostTriggerDuration=3000
&Event.E.ImageURLSettingsEnabled=yes
&Event.E.ImageURLSettings=streamprofile=MyProfile
```

Response

```
E3 OK
```

Step 3: Add an HTTP action of type upload to the event Event.E3. When the event is triggered, images will be uploaded to the HTTP server configured in step 1. Each time the event is triggered, the uploaded images will be saved to a time-stamped folder and named "image.<s>.jpg" where the sequence number <s> is generated automatically. (%25 is the URL code for the % character, %23 is the URL code for the # character)

```
http://myserver/axis-cgi/param.cgi?action=add
&group=Event.E3.Actions&template=httpaction
&Event.E3.Actions.A.Type=U
&Event.E3.Actions.A.Server=H0
&Event.E3.Actions.A.CreateFolder=MyFolder_%25F_%25H_%25M_%25S
&Event.E3.Actions.A.FileName=image.jpg
&Event.E3.Actions.A.Suffix=%23s
```

Response:

```
A0 OK
```

%F=the date in the format set with date.cgi (see VAPIX HTTP API specification)

%H=the hour as a decimal number (24 hour clock)

%M=the minute as a decimal number

%S=the second as a decimal number

#s=sequence number

(See *File Naming & Date/Time Formats* in the product's online help for more options.)

4 Event Handling and the HTTP API

4.1 Events

An event is a set of parameters that describes how and when the network video product should perform certain actions. Events can be scheduled to run during pre-programmed time periods, or triggered by some kind of signal, for example detected motion or a signal from a digital input port. Event actions specify what the network video product should do when an event occurs, for example upload images or activate output ports. An event may use an event server as destination for uploading images and notification messages.

Events, event actions and event servers are stored as dynamic parameter groups, i.e. the parameters are created at runtime. Please refer to *Development Guidelines* and *HTTP API 3.00* for an overview of dynamic parameters and the general parameter management CGI param.cgi.

The maximum number of event groups is limited if events are created from the product's web pages. If events are created with VAPIX using param.cgi, the force argument can be used to override this limit and create additional event groups. Avoid starting too many image buffers, however; the product may run out of memory.

Event triggers are specified by the parameters Event.E#.HWInputs (for hardware input triggers) and Event.E#.SWInput (for software input triggers).

Hardware input triggers include

- Input ports
- PIR sensor

Software input triggers include

- Manual trigger (see below)
- Motion detection
- Audio detection
- Camera tampering
- Temperature – when the temperature rises or falls outside the operating temperature of the product
- Video loss – the analog video signal is lost (applies to video encoders only)
- On boot, for example after power loss
- PTZ preset positions
- Network connectivity

Note: Supported triggers vary between product models.

The **manual trigger** is a virtual input connector which can be used to trigger any kind of action from an application. The manual trigger is also useful for testing purposes and can simulate activation or deactivation of any existing digital input. If the virtual input trigger is defined as

```
root.Event.E#.SWInput=IO<number>: /
```

use

```
http://<servername>/axis-cgi/io/virtualinput.cgi?action=<number+1>: /
```

to activate this virtual input. See also the HTTP API Specification.

Scheduled events run during pre-programmed time periods and are repeated every week. The recurrence pattern is defined by the Event.E#.Weekdays, Event.E#.Starttime and Event.E#.Duration parameters.

- If the event should run for more than 24 hours, specify one weekday only (e.g. Event.E#.Weekdays=1000000).
- To use two or more time periods of more than 24 hours, you must create two or more scheduled events (one for each period).
- To use different start times or durations for different days of the week, you must create separate events for each start time or duration.

4.2 Event actions

Event action specifies what the product should do when an event occurs. Each event can have one or more actions. The actions parameters are dynamic parameter groups created as a subgroup to an event (e.g. Event.E1.Actions.A0) and can be of different types:

- FTP upload (template fptaction)
- HTTP upload and notification (template httpaction)
- E-mail upload and notification (template smtpaction)
- TCP/IP notification (template tcpaction)
- Digital output control (template hwaction)
- Go to a preset Pan Tilt Zoom position (template ptzaction)
- Run a Guard Tour (template guardtouraction)
- Enable auto tracking (template autotrackingaction)
- Activate a light source (template lightcontrolaction)

Note: Supported actions are product dependent. See the product's Release notes for available templates.

The Event.E#.Actions.A# parameter group contains for example, information about the type of Action (notification or upload) and the event server to use. An action can be connected to one or more event servers and several actions can use the same event server.

Note: Image upload may affect the product's performance.

4.2.1 Dynamic event notification messages

Event notification messages, sent by e-mail or uploaded to a HTTP or TCP server, can contain modifier fields in addition to plain text. Modifier fields always start with a # or % character followed by another character and will be replaced with a value or a text string when the notification is sent.

The following modifier fields can be used:

- #e = The event name
- #E = The event description. See below.
- #w = The motion detection window name
- #t = The trigger type and value
- The modifier fields described in *File Naming & Date/Time Formats* in the product's online help. These modifiers can, for example, be used to time-stamp the notification message.

The default event descriptions (#E) listed in the table below are available in the file /etc/event_desc.list located in the product's internal file system. You can use the built-in editor

```
http://<servername>/admin-bin/editcgi.cgi?file=/etc/event_desc.list
```

to customize the descriptions.

Event type	#E	Notes
Video loss	Video loss on video %d	%d is replaced with the video source number starting from 1.
Boot	Restarting	
Manual	Manual trigger input port(s) %s went %s	The first %s is replaced with an integer, the second %s is replaced with "active" or "inactive".
Motion detection	Motion '%s' on motion window '%s'	The first %s is replaced with "started" or "stopped", the second %s is replaced with the window name.
Input port	Input port(s) %s went %s	The first %s is replaced with the port number starting from 1, the second %s is replaced with "active" or "inactive".
PTZ preset	Preset position '%s' on camera %d was reached	%s is replaced with the preset position number starting from 1, %d is replaced with the camera number starting from 1.

4.3 Event servers

An event server can be used to receive uploaded images (in MJPEG format) or event notification messages. Event servers are configured independently of events and several event actions may use the same event server. There are three types of event servers:

- **FTP servers** – Used for receiving uploaded images. Configured by the EventServers.FTP parameters.
- **HTTP servers** – Used for receiving uploaded or notification messages. Configured by the EventServers.HTTP parameters.
- **TCP/IP servers** – Used for receiving notification messages. Configured by the EventServers.TCP parameters.

In addition, **SMTP servers** can be used to send images and messages to specified e-mail addresses.

FTP, HTTP and TCP/IP servers are dynamic parameter groups within the EventServer group; SMTP (e-mail) server parameters are static and belong to the SMTP parameter group.

5 Parameters

Access control:

w=write, r=read; ad=administrator, op=operator, view=viewer

Example: w:op means that users with operator or administrator rights can write this parameter.

5.1 Events

An event is a set of parameters describing how and when the network video product should perform certain actions. Each event is defined by one parameter group.

Events can be triggered or scheduled. If an event is triggered, specify either HWInputs or SWInput.

[Event.E#]*

Template: event

Access Control – Create: operator

Access Control – Delete: operator

Group range: Product dependent. Can be overridden with the force argument (see param.cgi in the HTTP API specification).

Parameter	Default value	Valid values	Access control	Description
Name	New Event	A string	w:op r:op	Descriptive name of the event.
Type	T	S, T	w:op r:op	Event type. S = Scheduled T = Triggered
Enabled	yes	yes, no	w:op r:op	Enable/disable the event. Disabled events are never triggered.
Priority ¹	1	0, 1, 2	w:op r:op	Execution order. 0 = low 1 = normal 2 = high
HWInputs	xxxx ²	A string nnnn ² , where n={x,0,1,2}	w:op r:op	Specify a hardware input trigger. One character for each hardware input. ² x = do not trigger 0 = trigger on inactivation 1 = trigger on activation 2 = trigger on activation change A hardware input can, for example, be a digital input. Example 1: "1xxx" means trigger when hardware input 1 is activated. Example 2: "xx2x" means trigger when the state of hardware input 3 is

Parameter	Default value	Valid values	Access control	Description
				changed. Note: Cannot be combined with SWInput; use either HWInputs or SWInput
SWInput		A string	w:op r:op	Specify a software input trigger. See section 5.1.1 for more information. Note: Cannot be combined with SWInput; use either HWInputs or SWInput
Weekdays	1111111	A string	w:op r:op	Pattern of weekdays when the event should trigger. Example 1: 0111110 means trigger Monday – Friday. Example 2: 1000000 means trigger on Sundays. Note: If the event should run for more than 24 hours, specify one weekday only (e.g. 1000000).
Starttime	00:00	hh:mm	w:op r:op	Time when the event should start. Example 1: 07:30 means start at 7:30 am. Example 2: 19:30 means start at 7:30 pm.
Duration	24:00	hh:mm	w:op r:op	How long the event should be active. Max 168:00 hours (7 days).
MinimumTriggerInterval	00:00:00	hh:mm:ss	w:op r:op	Time interval between triggers; any triggers that occur during the interval are ignored. Max interval is 23:59:59. Example 1: 00:01:30 means that the interval between an event trigger and the next trigger for this event will be at least 1 minute and 30 seconds. Example 2: 00:00:00 disables the minimum trigger interval.
MinimumTriggerTimePeriod	00:00:00	hh:mm:ss	w:op r:op	The minimum time the trigger must be active before the event is triggered.
Image	0	An unsigned integer	w:op r:op	Image configuration (group Image.I#) associated with this event.
ImageURLSettingsEnabled	no	yes, no	w:op r:op	Use specific image settings for this event.

Parameter	Default value	Valid values	Access control	Description
ImageURLSettings		A string	w:op r:op	The images settings, for example a stream profile, to be used. Example: "streamprofile=MyProfile"
IncludePreTrigger	no	yes, no	w:op r:op	Use the pre-trigger buffer.
PreTriggerDuration	0	An unsigned integer	w:op r:op	Size of the pre-trigger buffer in milliseconds.
PreTriggerDurationUnit ³	s	s, m, h	w:op r:op	Unit for PreTriggerDuration in the web interface. s=second m=minute h=hour
IncludePostTrigger	no	yes, no	w:op r:op	Use the post-trigger buffer.
PostTriggerDuration	0	An unsigned integer	w:op r:op	Size of the post-trigger buffer in milliseconds.
PostTriggerDurationUnit ³	s	s, m, h	w:op r:op	Unit for PostTriggerDuration in the web interface. s=second m=minute h=hour
IncludeBestEffort	no	yes, no	w:op r:op	Enable best effort. Best effort is used for long continuous uploads.
BestEffortDuration	0	An unsigned integer	w:op r:op	Best effort duration in milliseconds. If IncludeBestEffort=yes and BestEffortDuration=0, images will be uploaded for as long as the event is active.
BestEffortDurationUnit ³	s	s, m, h	w:op r:op	Unit for BestEffortDuration in the web interface (available from plain config). s=second m=minute h=hour
MPEGPreTriggerDuration	0	An unsigned integer	w:op r:op	Size of the pre-trigger buffer for MPEG-4 and H.264 in milliseconds.
MPEGPostTriggerDuration	0	An unsigned integer	w:op r:op	Size of the post-trigger buffer for MPEG-4 and H.264 in milliseconds.
VideoFormat	jpeg	jpeg, mpeg4, h264	w:op r:op	Video format used for uploading/recording.

Parameter	Default value	Valid values	Access control	Description
IncludeAudio	no	yes, no	w:op r:op	Enable audio.
FrameRate	25fps	An integer (fps, fpm, fph)	w:op r:op	Frame rate per second, minute or hour. The same frame rate is used for the pre- and post-buffers. fps/fpm/fph=frames per second/minute/hour

¹ Nonfunctional at this time.

² The number of hardware inputs is product dependent.

³ These parameters only change how the value is presented in the web interface. The actual parameter value is always in the same unit.

* The # is replaced by an integer starting from zero, e.g. Event.E0

5.1.1 Software input triggers

The following software input trigger strings can be used. Support for software triggers vary between product models.

Trigger type	String	Description
Virtual input (manual trigger)	IO<number>: <state>	<number> is the virtual input number (an integer starting from 0). <state> can be: / or H = trigger when activated \ or L = trigger when inactivated Example: Use "IO0:H" to trigger when virtual input 1 is activated.
On boot	BOOT:/	Trigger on boot.
Video loss and resume (video encoders only)	V<source>: <trigger>	<source> can be: 0,1,... = video source number x = any video source <trigger> can be: \ = trigger on video loss / = ¹ trigger when video is resumed Example 1: "V0:\" means trigger when the video signal is lost on video source 1. Example 2: "Vx:\" means trigger when the video signal is lost on any video source.
Motion detection	M<number>: <state>	<number> is the motion detection window number (an integer starting from 0). <state> can be / = trigger when motion starts \ = trigger when motion stops x = trigger when motion starts or stops Example 1: "M1:/" means trigger when motion is detected in motion window 1.

Trigger type	String	Description
		Example 2: "M1:x" means trigger when motion starts or stops in motion window 1.
Audio detection	A<source>:<state>	<p><source> is the audio source number.</p> <p><state> can be:</p> <ul style="list-style-type: none"> / = trigger when the sound level rises above the alarm level. \ = trigger when the sound level falls below above the alarm level. x = trigger when the sound level passes the alarm level. <p>The alarm level is specified by the parameter Audio.A0.AlarmLevel.</p>
PTZ preset position	PP<preset>:<state>	<p><preset> can be:</p> <ul style="list-style-type: none"> • The PTZ preset position number, i.e. the last # in the PTZ.Preset.P#.Position.P# parameter • * = any preset position <p><state> can be:</p> <ul style="list-style-type: none"> / = trigger when the camera view arrives at the PTZ preset position \ =¹ trigger when the camera view leaves the PTZ preset position <p>Example 1: "PP1:/" means trigger when camera view comes to the preset position stored in PTZ.Preset.P0.Position.P1</p> <p>Example 2: "PP*:\ " means trigger when leaving any of the preset positions.</p>
IR cut filter	IR0:<state>	<p><state> can be:</p> <ul style="list-style-type: none"> \ = trigger when the IR cut filter is turned on / = trigger when the IR cut filter is turned off
Tampering	T<source>:/	<source> is the video source/camera number.
Temperature	TEMPO:/	Trigger when the temperature rises above or falls below the operating range of the product.
Do not trigger	"" (empty string)	Disable software input triggers.

¹ This option is not supported by the product's web interface. The trigger must be reconfigured from VAPIX if the event is modified from the web interface.

5.2 Event actions

Event actions specify what the product should do when events occur. An event can have multiple actions.

5.2.1 FTP Actions

The FTP action uploads images to an FTP server.

[Event.E#.Actions.A#]*

Template: ftpaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	U	U	w:op r:op	Type of action. U = Upload
Protocol	FTP	FTP	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Server		F0 ... Fn	w:op r:op	Primary FTP server ID. Refers to a parameter group under EventServers.FTP Example: "F0" refers to the parameter group EventServers.FTP.F0
Server2		F0 ... Fn	w:op r:op	Secondary FTP server ID.
CreateFolder		A string	w:op r:op	Upload path for the event. May contain modifier fields which will be expanded at the moment the event triggers. Example: "My_Folder_%F_%H_%M_%S" will create a folder named according to the date and time each time the event is triggered. See <i>File Naming & Date/Time Formats</i> in the product's online help pages for available modifiers.
FileName	image.jpg	A string	w:op r:op	Base file name for uploaded files. File name pattern: <basename>.<suffix>.<extension>

Parameter	Default value	Valid values	Access control	Description
Suffix	%y-%m-%d_%H-%M-%S-%f	A string	w:op r:op	Suffix to the base file name for uploaded files. Modifiers may be used in the suffix. <i>See File Naming & Date/Time Formats</i> in the product's online help pages for available modifiers.
MaxSequence Number	0	An unsigned integer	w:op r:op	The maximum value of when using a sequence number as file suffix. At this value the counter will wrap to 0.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.2 HTTP Actions

The HTTP action is used to send notification messages or upload images to an HTTP server.

[Event.E#.Actions.A#] *

Template: httpaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N, U	w:op r:op	Type of action. N = Notification U = Upload
Protocol	HTTP	HTTP	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Server		H0 ... Hn	w:op r:op	HTTP server ID. Refers to a parameter group under EventServers.HTTP Example: "H0" refers to the parameter group EventServers.HTTP.H0
Message		A string	w:op r:op	Used with Type=N. Notification message to send. Possible to use dynamic event notification, see 4.2.1.

Parameter	Default value	Valid values	Access control	Description
CustomParams		A string	w:op r:op	Used with Type=N. Additional CGI parameters to add to the URL to the HTTP server. The URL is defined in the parameter EventServers.HTTP.H#.Address Example: "foo=bar". Spaces are not allowed in this field and all text must be URI-encoded (see RFC3986). Example: To set the CGI parameter 'example' to 'Y' & 'Z', enter "example=Y+%26+Z" in this field.
CreateFolder		A string	w:op r:op	Used with Type=U. Upload path for the event. May contain modifier fields which will be expanded at the moment the event triggers. Example: "My_Folder_%F_%H_%M_%S" will create a folder named according to the date and time each time the event is triggered. See <i>File Naming & Date/Time Formats</i> in the product's online help pages for available modifiers.
FileName	image.jpg	A string	w:op r:op	Used with Type=U. Base file name for uploaded files. File name pattern: <basename>.<suffix>.<extension>
Suffix	%y-%m-%d_%H-%M-%S-%f	A string	w:op r:op	Used with Type=U Suffix to the base file name for uploaded files. Modifiers may be used in the suffix. See <i>File Naming & Date/Time Formats</i> in the product's online help pages for available modifiers.
MaxSequence Number	0	An unsigned integer	w:op r:op	Used with Type=U. The maximum value of when using a sequence number as file suffix. At this value the counter will wrap to 0.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.3 HW Actions

The hardware action activates equipment connected to the product's digital outputs.

[Event.E#.Actions.A#] *

Template: hwaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N = Notification
Protocol	HW	HW	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Output	1	1 ... <n> ²	w:op r:op	Output number to activate. Corresponds to the IOPort.I# parameter group.
Duration	0	An unsigned integer	w:op r:op	How long the output should be active in seconds. 0 = output is active for as long as the event is triggered.
Unit ³	s	s, m, h	w:op r:op	Unit for the Duration parameter in the web interface.

¹ Nonfunctional at this time.

² Number of outputs is product dependent. Check the product's Release notes.

³ This parameter only changes how the value is presented in the web interface. The actual parameter value is always in the same unit.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.4 SMTP Actions

The SMTP action is used to send notification message or uploaded images via e-mail.

[Event.E#.Actions.A#] *

Template: smtpaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N, U	w:op r:op	Type of action. N = Notification U = Upload
Protocol	SMTP	SMTP	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
EmailTo		A string	w:op r:op	E-mail address
Subject		A string	w:op r:op	Subject line. Possible to use dynamic event notification, see 4.2.1.
Message		A string	w:op r:op	Message body. Possible to use dynamic event notification, see 4.2.1.
ImagesPerMail	10	An unsigned integer	w:op r:op	Used with Type=U Maximum number of images to upload in one e-mail.
FileName	image.jpg	A string	w:op r:op	Used with Type=U Base file name for uploaded files. File name pattern: <basename>.<suffix>.<extension>
Suffix	%y-%m-%d_%H-%M-%S-%f	A string	w:op r:op	Used with Type=U Suffix to the base file name for uploaded files. Modifiers may be used in the suffix. See <i>File Naming & Date/Time Formats</i> in the product's online help pages for available options.

Parameter	Default value	Valid values	Access control	Description
MaxSequence Number	0	An unsigned integer	w:op r:op	Used with Type=U The maximum value of when using a sequence number as file suffix. At this value the counter will wrap to 0.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.5 TCP Actions

The TCP action is used to send notification messages via TCP/IP.

[Event.E#.Actions.A#] *

Template: tcpaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N = Notification
Protocol	TCP	TCP	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Server		T0 ... Tn	w:op r:op	TCP server ID. Refers to a parameter under EventServers.TCP Example: "T0" refers to the parameter group EventServers.TCP.T0
Message		A string	w:op r:op	Message body. Possible to use dynamic event notification, see 4.2.1.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.6 PTZ Actions

The PTZ action steers a PTZ camera to a PTZ preset position.

[Event.E#.Actions.A#] *

Template: ptzaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N = Notification
Protocol	PTZ	PTZ	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	1	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Server		P1 ... Pn	w:op r:op	Go to preset position P#. P# refers to the last part of the parameter PTZ.Preset.P#.Position.P# in which the first # is the image configuration used in the event, i.e. the value of Event.E#.Image Example: If Event.E#.Image=0 and Event.E#.Actions.A#.Server=P1, the camera will move to the position stored in PTZ.Preset.P0.Position.P1
Home	no	yes, no	w:op r:op	Return to the home position after the event.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.7 Guard tour actions

The Guard tour action starts a guard tour.

[Event.E#.Actions.A#] *

Template: guardtouraction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N = Notification
Protocol	GUARDTOUR	GUARDTOUR	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	0	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Name		G0 ... Gn	w:op r:op	Guard tour identification. Refers to the Guard Tour defined by the GuardTour.G# group.
Home	no	yes, no	w:op r:op	Return to the home position after the event.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.8 Auto tracking actions

The auto tracking action enables the camera's auto tracking functionality.

[Event.E#.Actions.A#] *

Template: autotrackingaction

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N = Notification
Protocol	AUTOTRACKING	AUTOTRACKING	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	0	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously).

Parameter	Default value	Valid values	Access control	Description
				Actions with different Order parameters will be executed sequentially in rising order.

¹ Nonfunctional at this time.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Actions.A0

5.2.9 Light control actions

The light control action activates a light source.

[Event.E#.Actions.A#]*

Template: lightcontrolaction

Access control – Create: operator

Access control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Type	N	N	w:op r:op	Type of action. N=Notification
Protocol	Light	Light	w:op r:op	Protocol. Describes the type of template used for this action.
Order ¹	0	0 ... 2	w:op r:op	Execution order. If an event has multiple actions, actions with the same Order will be executed in parallel (simultaneously). Actions with different Order parameters will be executed sequentially in rising order.
Light	1	1	w:op r:op	The light source this action controls.
Activate	-100	<action>[, <action>] ²	w:op r:op	The desired activation action(s).
Inactivate	-0	<action>[, <action>] ²	w:op r:op	The desired inactivation action(s).
Duration	0	0 ...	w:op r:op	The time between the Activate and Inactivate actions. See also Unit below.
Unit	s	s, h	w:op r:op	Unit for the Duration parameter s = seconds h = 1/100 seconds

¹ Nonfunctional at this time.

² Please refer to VAPIX® Light Control API for valid action strings.

* The # is replaced by an integer starting from zero, e.g. Event.E0.Action.A0.

5.3 Event servers

5.3.1 FTP Servers

FTP servers can receive uploaded images.

[EventServers.FTP.F#]*

Template: ftp_config

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Name	NewFTPServer	A string	w:op r:op	A descriptive name.
Address		An IP address or a host name	w:op r:op	IP address or host name of the server
Login		A string	w:op r:op	FTP user name
Password		A string	w:op r:op	FTP password
UploadPath		A string	w:op r:op	Directory where uploaded files go
Port	21	0 ... 65535	w:op r:op	FTP port
Passive	no	yes, no	w:op r:op	Enable/disable passive FTP
UseTempFile	no	yes, no	w:op r:op	Some FTP servers do not allow an existing file to be overwritten by a new file with the same name. When this option is enabled, the new file will be uploaded with a temporary name. The original file is then deleted and the new file is given the original file name.

* The # is replaced by an integer starting from zero, e.g. EventServers.FTP.F0

5.3.2 HTTP Servers

HTTP servers can receive notification messages and uploaded images.

[EventServers.HTTP.H#] *

Template: http_config

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Name	NewHTTPServer	A string	w:op r:op	A descriptive name.
Address		A string	w:op r:op	URL to the server, including name of CGI script to handle the request. Example: "http://192.168.254.10/cgi-bin/upload.cgi"
Login		A string	w:op r:op	HTTP user name
Password		A string	w:op r:op	HTTP password
Proxy		An IP address	w:op r:op	Proxy address
ProxyPort		0 ... 65535	w:op r:op	Proxy port number
ProxyLogin		A string	w:op r:op	Proxy user name
ProxyPassword		A string	w:op r:op	Proxy password

* The # is replaced by an integer starting from zero, e.g. EventServers.HTTP.H0

5.3.3 TCP Servers

TCP servers can receive TCP/IP notification messages.

[EventServers.TCP.T#]*

Template: tcp_config

Access Control – Create: operator

Access Control – Delete: operator

Parameter	Default value	Valid values	Access control	Description
Name	NewTCPServer	A string	w:op r:op	A descriptive name.
Address		An IP address or a host name	w:op r:op	IP address or host name of the server
Port		0 ... 65535	w:op r:op	TCP port number

* The # is replaced by an integer starting from zero, e.g. EventServers.TCP.T0

5.4 SMTP Servers

SMTP servers are used when sending e-mail messages from the network video product.

[SMTP]

Parameter	Default value	Valid values	Access control	Description
FromEmail		An e-mail address	w:ad r:ad	E-mail address. The e-mail address can belong to an individual person or to the Axis device from which the e-mail is sent.
MailServer1		An IP address or a host name	w:ad r:ad	Primary mail server
MailServer2		An IP address or a host name	w:ad r:ad	Secondary mail server

5.4.1 SMTP Authentication

[SMTP.Authentication.A#]*

Parameter	Default value	Valid values	Access control	Description
Enabled	no	yes, no	w:ad r:ad	Enable or disable authentication.
UserName		A string	w:ad r:ad	The user name for the mail server or the POP server.

Parameter	Default value	Valid values	Access control	Description
Password		A string	w:ad r:ad	The password for the mail server or the POP server.
WeakestMethod	Login	Login, Plain, CRAM-MD5, DIGEST-MD5	w:ad r:ad	The weakest method allowed for mail server login.
UsePOP	no	yes, no		Login to a POP server.
POPServer		An IP address or a host name	w:ad r:ad	POP server.

* The # is replaced by a group number 1 or 2 representing mail servers 1 and 2, respectively. For example, SMTP.Authentication.A1

6 References

All VAPIX references are available at
<http://www.axis.com/vapix/>