# VAPIX® version 3

## I/O Port API

AXIS COMMUNICATIONS

# VAPIX LICENSE AGREEMENT

This is a legal agreement (the "License Agreement") between you (either individual or an entity) and Axis Communications AB (hereinafter referred to as "Axis").

1. GRANT OF LICENSE

   Axis hereby grants to you the right to use the INTERFACE and 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 cameras, Axis video servers, Axis video encoders and Axis video decoders (as defined by Axis at its discretion) 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 the INTERFACE DESCRIPTION outside the limited purpose set forth in Section 1 above is strictly prohibited.

3. NO REVERSE ENGINEERING

   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. 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 THE INTERFACE DESCRIPTION whether for the purpose set forth in Section 1 above or not.

5. 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.

6. DISCLAIMER

6.1. 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.

6.2. YOU ARE YOURSELF RESPONSIBLE FOR EXAMINING WHETHER THE INTERFACE AND THE INTERFACE DESCRIPTION ARE ENCUMBERED BY OR INFRINGES UPON A RIGHT HELD BY A THIRD PARTY. AXIS, WHO HAS NOT UNDERTAKEN ANY SUCH INVESTIGATIONS, HAS NO KNOWLEDGE OF NOR DOES AXIS ACCEPT ANY LIABILITY FOR ANY SUCH ENCUMBRANCES OR INFRINGEMENTS.

6.3. YOU UNDERTAKE NOT TO PURSUE ANY CLAIMS WHATSOEVER AGAINST AXIS OR ITS AFFILIATES RELATING TO OR EMANATING FROM THE INTERFACE AND THE INTERFACE DESCRIPTION.

6.4. 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.

6.5. 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.

6.6. YOU SHALL INDEMNIFY AND HOLD AXIS AND ITS AFFILIATES HARMLESS FROM ANY CLAIMS WHATSOEVER FROM ANY THIRD PARTY AGAINST AXIS OR ITS AFFILIATES RELATING TO OR EMANATING FROM YOUR USE OF THE INTERFACE AND THE INTERFACE DESCRIPTION UNDER THIS LICENSE AGREEMENT. THE FOREGOING INDEMNIFICATION INCLUDES BUT IS NOT LIMITED TO ANY AND ALL DAMAGES, COSTS AND EXPENSES (INCLUDING REASONABLE ATTORNEYS' FEES).

# Table of Contents

# 1 Overview

## 1.1 Description

Most of Axis network cameras and video encoders have integrated digital input and output ports which enable connection to external devices such as detectors, lights, switches and alarm relays. The number of I/O ports is product dependent. In some products, each I/O port can be configured to act as input or output.

Input ports – For external alarm devices that can toggle between an open and grounded (closed) circuit, for example a pushbutton or a door sensor. Devices connected to input ports are usually used to trigger alarms.

Output port – For external alarm devices such as relays and LEDs that, for example, are to be activated by a triggered/scheduled event.

### Important!

**Numbering of I/O ports**

The numbering of the I/O ports starts from ***one*** in port.cgi requests and in all responses except in the response to the monitor request. In the response to the monitor request, port numbering starts from ***zero***. For the IOPort.I# parameter groups port numbering starts from ***zero***.

The physical port labeled *n* is thus *n* in port.cgi requests, but by *n-1* in the response to the monitor request. The corresponding parameter group is IOPort.I(*n-1*).

The port.cgi described in this document (see section 5) replaces the input.cgi and output.cgi from VAPIX version 2. The input.cgi and output.cgi are obsolete but supported for backwards compatibility. For products with configurable ports, port.cgi must be used.

## 1.2 History

| Version | Date | Comment |
|---|---|---|
| 1.00 | 2008-11-19 | Initial version. |

# 2 Prerequisites

## 2.1 Identification

**Property:** Properties.API.Version=3

Port *n+1* is configurable if IOPort.I*n*.Configurable=yes (*n* is a non-negative integer)

**Firmware:** 5.00 and above

**Product category:** Products with I/O connectors.

## 2.2 Dependencies

The parameters (see section 4) are managed through the general parameter handling CGI param.cgi. Parameter values can be listed and updated. See the VAPIX® HTTP API documentation for more information.

# 3 Common Examples

**Numbering of I/O ports**

I/O port numbering starts from ***one*** in port.cgi requests and in all responses except in the response to the monitor request. In the response to the monitor request, port numbering starts from ***zero***. See example 2 below. For the IOPort.I# parameter groups port numbering starts from ***zero***.

In example descriptions port *n* refers to the physical port labeled *n*. This port is thus *n* in port.cgi requests, but *n-1* in the monitor response. The corresponding parameter group is IOPort.I(*n-1*).

**Example 1:** Check whether port 3 is active.

```
http://myserver/axis-cgi/io/port.cgi?checkactive=3
```

Response
HTTP Code: 200 OK
Content-Type: text/plain
Body:

```
port3=inactive
```

**Example 2:** Monitor data on port 3. Note how port numbering starts from zero in the response.

```
http://myserver/axis-cgi/io/port.cgi?monitor=3
```

**a)** Port 3 is an input port.

Response:
HTTP Code: 200 OK
Content-Type: multipart/x-mixed-replace; boundary=ioboundary
Body:

```
--ioboundary
Content-Type: text/plain

2I:/
```

```
--ioboundary
Content-Type: text/plain

2I:H

--ioboundary
Content-Type: text/plain

2I:\

--ioboundary
Content-Type: text/plain

2I:L

--ioboundary
Content-Type: text/plain



--ioboundary
Content-Type: text/plain

 ...
```

**Note**: Non-empty boundaries are sent when the port status changes. If there are no changes, empty boundaries are sent at 15-second intervals.

**b)** Port 3 is an output port.

Response:
HTTP Code: 200 OK
Content-Type: multipart/x-mixed-replace; boundary=ioboundary
Body:

```
--ioboundary
Content-Type: text/plain

2O:/

--ioboundary
Content-Type: text/plain

2O:H

--ioboundary
Content-Type: text/plain

2O:\

--ioboundary
Content-Type: text/plain

2O:L

--ioboundary
Content-Type: text/plain


```

```
--ioboundary
Content-Type: text/plain

 ...
```

**Example 3:** Check the direction of ports 1 and 2

```
http://myserver/axis-cgi/io/port.cgi?checkdirection=1,2
```

Response
HTTP Code: 200 OK
Content-Type: text/plain
Body:

```
port1=input
port2=output
```

**Example 4:** Set port 2 to active (Only valid if the port is configured as output).

```
http://myserver/axis-cgi/io/port.cgi?action=2:/
```

**Example 5:** For port 2, set two 300 ms pulses with 500 ms delay between the pulses. (Only valid if port 2 is configured as output.)

```
http://myserver/axis-cgi/io/port.cgi?action=2:/300\500/300\
```

**Example 6:** Retrieve information about port 1.

```
http://myserver/axis-cgi/param.cgi?action=list&group=IOPort.I0
```

Response:
HTTP Code: 200 OK
Content-Type: text/plain
Body:

```
root.IOPort.I0.Configurable=yes
root.IOPort.I0.Direction=output
root.IOPort.I0.Input.Name=Input 1
root.IOPort.I0.Input.Trig=closed
root.IOPort.I0.Output.Name=Output 1
root.IOPort.I0.Output.Active=open
root.IOPort.I0.Output.Button=actinact
root.IOPort.I0.Output.PulseTime=0
```

**Example 7:** Configure port 2 to act as output. This example is only applicable to configurable ports.

```
http://myserver/axis-cgi/param.cgi?action=update&
IOPort.I1.Direction=output
```

# 4 I/O Parameters

**Security level note**: c=create, d=delete, w=write (set), r=read (get). For example, operator:rw means that the operator has read and write access.

## 4.1 Port settings

The settings for the physical I/O port numbered # are defined by the group IOPort.I#, where # is an integer starting from 0 (zero) for the first physical port.

**[IOPort.I#]***

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| Configurable | yes[1] | yes,<br>no | admin:r<br>operator:r | The port is configurable or not. |
| Direction | input | input,<br>output | admin:rw<br>operator:r | The port is configured to act as input or output. Read-only for non-configurable ports. |
| Input.Name[2] | Input 1 | A string | admin:rw<br>operator:r | User-friendly name for the input. |
| Input.Trig[2] | closed | closed,<br>open | admin:rw<br>operator:r | Determines when to trig. |
| Output.Name[3] | Output 1 | A string | admin:rw<br>operator:r | User-friendly name for the output. |
| Output.Active[3] | closed | closed,<br>open | admin:rw<br>operator:r | The active state of the output. |
| Output.Button[3] | none | none,<br>pulse,<br>actinact | admin:rw<br>operator:r | The button type associated with this output. If enabled, the Output button is displayed on the Live View page.<br>actinact = Active/Inactive |
| Output.PulseTime[3] | 0 | An integer | admin:rw<br>operator:r | The pulse time for the pulse button connected with this output, if any. |

[1] Product/release dependent. See the product's Release notes.
[2] Not supported for non-configurable output ports.
[3] Not supported for non-configurable input ports.

* # represents a physical port and should be replaced by an integer starting from zero. The number of physical ports is product dependent.

# 5 HTTP API – port.cgi

## 5.1 Request

Retrieve information about port status and directions, activate/deactivate and monitor ports.

**Port numbering**
In port.cgi requests and in all responses *except* the monitor response, port numbering (*id* below) starts from ***one*** (where one corresponds to the physical port labeled '1'). In the *monitor response* port numbering starts from ***zero***.

**Security level:** viewer
**Method**: GET
**Syntax:**

```
http://<servername>/axis-cgi/io/port.cgi?
<argument>=<value>[&<argument>=<value>...]
```

with the following arguments and values

| Argument | Valid values | Description |
|---|---|---|
| check=<*int*>[,<*int*>,...] | <*id1*>[,<*id2*>,...] | Return the status (1 or 0) of one or more ports numbered <*id1*>,<*id2*>, ...<br>1=active, 0=inactive |
| checkactive=<*int*>[,<*int*>,...] | <*id1*>[,<*id2*>,...] | Return the status (active or inactive) of one or more ports numbered <*id1*>, <*id2*>, ... This value depends on the parameters Output.Active for an output and Input.Trig for an input. If the port is an output and Output.Active is configured as closed, then this request will return active if the port state is closed. |
| checkdirection=<*int*>[,<*int*>,...] | <*id1*>[,<*id2*>,...] | Return the port direction (input or output) of one or more ports numbered <*id1*>, < *id2*>, ... |
| monitor=<*int*>[,<*int*>,...] | <*id1*>[,<*id2*>,...] | Return a multipart stream of "check" ports (see return description below). Input and output ports must be monitored separately. |
| action=<*string*>[1] | [<*id*>]:<*a*>[<*wait*><*a*>...] | Set the port relay <*id*> active or inactive and wait <*wait*> milliseconds.<br><br><*id*> = Port number. Default: Output 1<br><*a*> = Action character: / or \<br>/=active, \=inactive<br><*wait*> = delay in milliseconds<br>Example: 1:/ sets output 1 active |

[1] Valid for output ports only.

## 5.2 Responses

### 1. Success, all arguments except monitor

**Return:**
HTTP Code: 200 OK
Content-Type: text/plain
Body:

```
port<id>=<information>
```

Note: The body is empty for the action argument.

### 2. Success, argument monitor

**Return:**
HTTP Code: 200 OK
Content-Type: multipart/x-mixed-replace; boundary=<*boundary*>
Body:

```
--<boundary>
<monitor data>

where the returned <monitor data> is

Content-Type: text\plain

<id><port>:<action character>

--<boundary>
<monitor data>
```

Here <id> is the port number and <port> is I for inputs and O for outputs. The action character is / or H for active and \ or L for inactive ports.

**Note**: Non-empty boundaries are sent when the port status changes. If there are no changes, empty boundaries are sent at 15-second intervals.

### 3. Failure – Server error

Error during execution.

HTTP Code: 500 Internal Server Error
Content-Type: text/plain
Body:

```
<body>
```

### 4. Failure – Bad request

The request was malformed or not implemented.

HTTP Code: 400 Bad Request
Content-Type: text/plain
Body:

```
<body>
```

### 5. Failure – Method not allowed

The request is not valid.

HTTP Code: 405 Method Not Allowed
Content-Type: text/plain
Body:

```
<body>
```

# 6 References

All VAPIX references are available at
http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php

VAPIX® HTTP API version 3

VAPIX® Parameter Specification