VAPIX® VERSION 3

Pan Tilt Zoom API

AXIS COMMUNICATIONS

COPYRIGHT NOTICE

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 the 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 THE 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.

Pan Tilt Zoom API

Table of Contents

1 Overview 5
1.1 Description 5
1.1.1 PTZ Parameter and CGI Functionality 5
1.2 Definitions 5
1.3 References 5
2 Common Examples 6
3 PTZ Control API 8
3.1 Description 8
3.2 Prerequisites 8
3.2.1 Identification 8
3.3 Parameters 8
3.3.1 PTZ 8
3.3.2 PTZ.ImageSource 9
3.3.3 PTZ.PTZDriverStatuses 9
3.3.4 PTZ.Support 10
3.3.5 PTZ.Limit 13
3.3.6 PTZ.Various 14
3.3.7 PTZ.UserBasic 17
3.3.8 PTZ.UserAdv 17
3.4 HTTP API 20
3.4.1 PTZ Control 20
3.4.2 PTZ Configuration 23
3.4.3 Response 24
4 PTZ Driver Management API 28
4.1 Description 28
4.2 Prerequisites 28
4.2.1 Identification 28
4.3 Parameters 28
4.3.1 PTZ.CamPorts 28
4.4 HTTP API 29
4.4.1 PTZ Driver Upload 29
4.4.2 PTZ Driver Installation 31
4.4.3 Response 32
5 Queuing API 33
5.1 Description 33
5.2 Prerequisites 33
5.2.1 Identification 33
5.3 Parameters 33
5.3.1 PTZ.Various 33
5.3.2 PTZ.UserCtlQueue.U# 34
5.4 HTTP API 35
5.4.1 PTZ Control Queue 35
5.4.2 Response 36

Pan Tilt Zoom API

1 Overview

1.1 Description

This document describes the Pan/Tilt/Zoom (PTZ) functionality in VAPIX®. The PTZ API is used for controlling PTZ functionality on Axis PTZ dome cameras, video encoders, cameras with digital PTZ, and fixed cameras with serial ports. PTZ can be controlled for each video channel (referred to as a camera in the API) on an Axis product. The PTZ API is divided in PTZ Control API, PTZ Driver Management API and Queuing API. This document also describes the parameters HTTP API for IR cut filter, backlight compensation, focus, brightness, iris, auxiliary and OSD menu.

1.1.1 PTZ Parameter and CGI Functionality

Each functionality (for example pan) uses `Support` parameters (see *3.3.4 PTZ.Support, on page 10*) to check if the functionality is supported, uses `Various` parameters (see *3.3.6 PTZ.Various*) to check if functionality is enabled and to disable functionality, and use CGIs (see *3.4.1 PTZ Control*) to control the functionality.

1.2 Definitions

Term	Definition
Digital PTZ	PTZ without moving parts. PTZ "movements" are done by dynamically changing crop section on the image sensor.
Mechanical PTZ	Mechanical PTZ is driven by motors. The motors could either be integrated in the Axis product or the Axis product could be mounted on a external mechanical PT (PanTilt) device. It could also be an external PTZ camera connected to an encoder.
Panopsis technology	Panopsis technology is used by mechanical PTZ domes that in addition to the standard zoom lens have a Panopsis (fisheye) lens attached to the dome cover. A product with Panopsis technology can be used in two modes. In Overview mode, the zoom lens is fully zoomed out and points straight down (-90 degrees tilt) looking through the Panopsis lens. This gives a 360-degree overview image. In Normal mode, the zoom lens avoids the Panopsis lens and the product works as a standard mechanical PTZ dome. Note that some PTZ commands cannot be used or do not have any effect in Overview mode.
Absolute movement	Move to a specific coordinate position, for example "move to -45°".
Relative movement	Move a specific distance from the current position, for example "move left 15°".
Continuous movement	Move in a specific direction (until stopped), for example "move down".
Auto-flip	The auto-flip functionality allows the camera to simulate a continuous pan beyond the mechanical stop, thereby enabling an operator to continuously follow an object.
Static PTZ driver	A PTZ driver is built into the firmware and is always running. Typically used on domes and digital PTZ.
Uploadable PTZ driver	A PTZ driver that can be uploaded and installed to a running Axis product.

1.3 References

All VAPIX® references are available at:

http://www.axis.com/vapix

2 Common Examples

Example 1:
Check if PTZ is available.

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

Example 2:
Verify that PTZ is enabled on channel 1:

```
http://myserver/axis-cgi/com/ptz.cgi?info=1&camera=1
```

A successful response should look similar or the same to the response in *3.4.3.3 Success for info=1, on page 25*.

Example 3:
Request information about which PTZ commands that are available for `camera=1`. For the response see example in section *3.4.1 PTZ Control*.

```
http://myserver/axis-cgi/com/ptz.cgi?info=1&camera=1
```

Example 4:
Pan `camera=3` to the right, 10 degrees.

```
http://myserver/axis-cgi/com/ptz.cgi?rpan=10&camera=3
```

Example 5:
Set a home position named "MyPreset" at the current location for the Axis product.

```
http://myserver/axis-cgi/com/ptzconfig.cgi?setserverpresetname=MyPreset&home=yes
```

Example 6:
Enable PTZ on video channel number 3 on an Axis product with digital PTZ. For digital PTZ, `PTZ.ImageSource.I0.PTZEnabled` is controlling all video channel. Each video channel has a `PTZ.Various.V#.Locked` parameter.

```
http://myserver/axis-cgi/param.cgi?PTZ.ImageSource.I0.PTZEnabled=yes&PTZ.Vari-
ous.V3.Locked=false
```

Example 7:
Check if a product supports upload and installation of drivers.

```
http://myserver/axis-cgi/param.cgi?action=list&group=Properties.PTZ.DriverManagement
```

Example 8:
Upload a driver with the driver file contents as HTTP POST data (see *page 29* for details).

```
http://myserver/axis-cgi/ptz/ptzuploader.cgi
```

A driver ID is returned in the HTTP response, for example `7`.

```
<?xml version=1"1.0"?>
<root>
 <id>7</id>
</root>
```

Example 9:

Install the uploaded driver with ID 7 on video channel 3.

```
http://myserver/axis-cgi/ptz/ptzupgrader.cgi?driverid=7&channel=3
```

3 PTZ Control API

3.1 Description

There are two types of PTZ, mechanical PTZ and digital PTZ. Mechanical PTZ is driven by motors. The motors could either be integrated in the Axis product or the Axis product could be mounted on a external mechanical PT (PanTilt) device. It could also be an external PTZ camera connected to an encoder. For Axis products with digital PTZ it is possible to select only a part of the image by zooming in. The zoomed area can then be moved around using pan/tilt commands. Digital PTZ uses the same HTTP API as Axis products with mechanical PTZ, with some limitations.

The PTZ Management API consists of the following CGIs:

`ptz.cgi`	Control the PTZ movement.
`ptzconfig.cgi`	Configure the PTZ functionality.
`ptzqueue.cgi`	Control the PTZ control queuing mechanism.

3.2 Prerequisites

3.2.1 Identification

Property: `Properties.API.HTTP.Version=3`
Property: `Properties.PTZ.PTZ=yes`
Property: `Properties.PTZ.DigitalPTZ=yes` (Products with digital PTZ)
Property: `Properties.Overview.Overview=yes` (Products with Panopsis technology)
Property: `Properties.Overview.MechanicalHybrid=yes` (Products with Panopsis technology)
Firmware: 5.20 and later.

3.3 Parameters

3.3.1 PTZ

Various Pan Tilt Zoom parameters. All writable parameters may be directly modified.

[PTZ]

Parameter	Default values	Valid values	Access control	Description
`BoaProtPTZOperator`	`password`	`password` `anonymous`	admin: read, write operator: read viewer: read	`password` = Password is required to control PTZ units and to use control queue. `anonymous` = Anybody on the network have access to PTZ units without having to log in.
`CameraDefault`	`1`	`1 ... n` [1]	admin: read, write operator: read viewer: read	The video channel used if the `camera` parameter is omitted in HTTP requests.
`NbrOfCameras`	Product specific	`1 ...`	admin: read operator: read	Number of video channels.
`NbrOfSerPorts`	Product specific	`0 ...`	admin: read operator: read viewer: read	Number of serial ports.

[PTZ] (Continued)

1. n = Number of video channels.

3.3.2 PTZ.ImageSource

This group contains a parameter for turning on/off PTZ functionality.

Toggling this parameter does not install/uninstall any driver.

[PTZ.ImageSource.I#]

Parameter	Default values	Valid values	Access control	Description
`PTZEnabled`	**Mechanical PTZ**: `true` **Digital PTZ**: `false`	`true` `false`	admin: read, write operator: read, write viewer: read	**Mechanical PTZ** = Enable/disable the PTZ functionality of the video channel. `true` = Enable the PTZ functionality of the video channel. `false` = Disable the PTZ functionality of the video channel. **Digital PTZ** = Enable/disable the PTZ functionality on the Axis product. `true` = Enable the PTZ functionality on the Axis product. `false` = Disable the PTZ functionality on the Axis product. When this parameter is changed to `false`, the product returns to a default PTZ position (in contrast to the parameter `PTZ.Various.V#.Locked`, for which the position is locked when the parameter is set to `true`).

Note

For mechanical PTZ, the index # in `PTZ.ImageSource.I#` is the index of the video channel, starting on `0` for video channel 1. For digital PTZ, there is one group `PTZ.ImageSource.I0`, and the parameter `PTZ.ImageSource.I0.PTZEnabled` is used for enabling/disabling PTZ on all channels.

3.3.3 PTZ.PTZDriverStatuses

This group contains a parameter that informs about the current status of the drivers installed on the Axis product.

[PTZ.PTZDriverStatuses]

Parameter	Default values	Valid values	Access control	Description
Driver#Status		0 ... 3	admin: read operator: read viewer: read	The status of the driver installed on camera # , where # starts on 1 for video channel 1. The value should be interpreted as: 0 = No driver installed. 1 = Installation file invalid or incompatible. 2 = Driver malfunction. 3 = Driver installed.

3.3.4 PTZ.Support

The dynamic parameter group PTZ.Support.S# is updated when a driver is installed on a video channel. A parameter in the group has the value true if the corresponding capability is supported by the driver. The index # is the video channel number which starts from 1.

An absolute operation means moving to a certain position, a relative operation means moving relative to the current position. Arguments referred to apply to *3.4.1 PTZ Control*.

[PTZ.Support.S#]

Parameter	Default values	Valid values	Access control	Description
AbsolutePan	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute pan is supported by the driver. false = Absolute pan is disabled/not supported.
RelativePan	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative pan is supported by the driver.[1] false = Relative pan is disabled/not supported.
AbsoluteTilt	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute tilt is supported by the driver. false = Absolute tilt is disabled/not supported.
RelativeTilt	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative tilt is supported by the driver.[1] false = Relative tilt is disabled/not supported.
AbsoluteZoom	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute zoom is supported by the driver.[1] false = Absolute zoom is disabled/not supported.
RelativeZoom	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative zoom is supported by the driver.[1] false = Relative zoom is disabled/not supported.

[PTZ.Support.S#] (Continued)

DigitalZoom	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Digital zoom is supported (increases the upper limit of PTZ.Limit.L#.MaxZoom to 19999) by the driver.[1] false = Digital zoom is disabled/not supported.
AreaZoom	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Area zoom is supported by the driver.[1] false = Area zoom is disabled/not supported.
AbsoluteFocus	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute focus is supported by the driver.[1] false = Absolute focus is disabled/not supported.
RelativeFocus	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative focus is supported by the driver.[1] false = Relative focus is disabled/not supported.
AutoFocus	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Autofocus is supported by the driver. false = Autofocus is disabled/not supported.
AbsoluteIris	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute iris is supported by the driver. false = Absolute iris is disabled/not supported.
RelativeIris	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative iris is supported by the driver. false = Relative iris is disabled/not supported.
AutoIris	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Auto iris is supported by the driver. false = Auto iris is disabled/not supported.
AbsoluteBright-ness	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Absolute brightness is supported by the driver. false = Absolute brightness is disabled/not supported.
RelativeBright-ness	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Relative brightness is supported by the driver. false = Relative brightness is disabled/not supported.
ContinuousPan	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Continuous pan is supported by the driver. false = Continuous pan is disabled/not supported.
ContinuousTilt	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Continuous tilt is supported by the driver. false = Continuous tilt is disabled/not supported.

[PTZ.Support.S#] (Continued)

`ContinuousZoom`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Continuous zoom is supported by the driver.[1] `false` = Continuous zoom is disabled/not supported.
`ContinuousFocus`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Continuous focus is supported by the driver.[1] `false` = Continuous focus is disabled/not supported.
`ContinuousIris`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Continuous iris is supported by the driver. `false` = Continuous iris is disabled/not supported.
`ContinuousBrig-htness`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Continuous brightness is supported by the driver. `false` = Continuous brightness is disabled/not supported.
`Auxiliary`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = CGI argument `auxiliary` (`com/ptz.cgi?auxiliary=<string>`)is supported by the driver. `false` = CGI argument `auxiliary` is disabled/not supported.
`ServerPreset`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Server presets are supported by the driver. `false` = Server presets are disabled/not supported.
`DevicePreset`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Device presets are supported by the driver. `false` = Device preset are disabled/not supported.
`SpeedCtl`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Speed control is supported by the driver. `false` = Speed control is disabled/not supported.
`IrCutFilter`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = IR cut filter is supported by the driver. `false` = IR cut filter is disabled/not supported.
`AutoIrCutFilter`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = The IR cut filter can be controlled automatically. `false` = Automatic IR cut filter is disabled/not supported.
`Backlight`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Backlight compensation is supported by the driver. `false` = Backlight compensation is disabled/not supported.

[PTZ.Support.S#] (Continued)

OSDMenu	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = OSD menu is supported by the driver. false = OSD menu is disabled/not supported.
ActionNotification	Product/driver dependent	true false	admin: read, write operator: read viewer: read	Value is true if the PTZ driver can send messages to other internal applications when it starts or stops movements. Makes it possible to trigger events on arrival to a preset position. true = Action notification is supported. false = Action notification is disabled/not supported.
ProportionalSpeed	Product/driver dependent	true false	admin: read, write operator: read viewer: read	Value is true if the product supports proportional speed when using the command continuouspantiltmove, for example adjusting the movement speed in the image proportional to the zoom level used. true = Proportional speed is supported. false = Proportional speed is disabled/not supported.
LensOffset	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Lens offset is supported by the driver. false = Lens offset is disabled/not supported.

1. Products with Panopsis technology: Even if supported, functionality cannot be used in Overview mode.

3.3.5 PTZ.Limit

This dynamic group is updated when a driver is installed on a video channel. Index # is the video channel number, starting on 1. When it is possible to obtain the current position from the driver, for example the current pan position, it is possible to apply limit restrictions to the requested operation. For instance, if an absolute pan to position 150 is requested, but the upper limit is set to 140, the new pan position will be 140. This is the purpose of all but MinFieldAngle and MaxFieldAngle in this group. The purpose of those two parameters is to calibrate image centering.

[PTZ.Limit.L#]

Parameter	Default values	Valid values[1]	Access control	Description
MinPan	Product/driver dependent	-180 ... 180	admin: read, write operator: read viewer: read	Lower limit for pan position.
MaxPan	Product/driver dependent	-180 ... 180	admin: read, write operator: read viewer: read	Upper limit for pan position.

[PTZ.Limit.L#] (Continued)

MinTilt	Product/driver dependent	-180 ... 180	admin: read, write operator: read viewer: read	Lower limit for tilt position.
MaxTilt	Product/driver dependent	-180 ... 180	admin: read, write operator: read viewer: read	Upper limit for tilt position.
MinZoom	Product/driver dependent	1 ... 9999[2]	admin: read, write operator: read viewer: read	Lower limit for zoom position.
MaxZoom	Product/driver dependent	1 ... 9999[2]	admin: read, write operator: read viewer: read	Upper limit for zoom position.
MinFocus	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Lower limit for focus position.
MaxFocus	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Upper limit for focus position.
MinIris	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Lower limit for iris position.
MaxIris	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Upper limit for iris position.
MinBrightness	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Lower limit for brightness position.
MaxBrightness	Product/driver dependent	1 ... 9999	admin: read, write operator: read viewer: read	Upper limit for brightness position.
MinFieldAngle	Product/driver dependent	1 ... 1000	admin: read, write operator: read viewer: read	Minimum field angle for the (zoom) lens, used to calibrate image centering.
MaxFieldAngle	Product/driver dependent	1 ... 1000	admin: read, write operator: read viewer: read	Maximum field angle for the (zoom) lens, used to calibrate image centering.

1. May be overridden by configuration file for driver. Maximal value must be ³ minimal value.
2. If digital zoom is supported (see PTZ.Support.S#.DigitalZoom in section *3.3.4 PTZ.Support*), MaxZoom can have values up to 19999.

3.3.6 PTZ.Various

The dynamic parameter group `PTZ.Various.V#` is updated when a driver is installed on a video channel. The index `#` is the video channel number which starts from `1`.

The group consists of several different types of parameters for the video channel. To distinguish the parameter types, the group is presented as three different categories below.

The `Enabled` parameters determine if a specific feature can be controlled using `ptz.cgi` (see section *3.4.1 PTZ Control*).

[PTZ.Various.V#]

Parameter	Default values	Valid values[1]	Access control	Description
`PanEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Pan is allowed. `false` = Pan is not allowed.
`TiltEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Tilt is allowed. `false` = Tilt is not allowed.
`ZoomEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Zoom is allowed.[2] `false` = Zoom is not allowed.
`FocusEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Focus is allowed.[2] `false` = Focus is not allowed.
`IrisEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Iris is allowed. `false` = Iris is not allowed.
`BrightnessEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Brightness is allowed. `false` = Brightness is not allowed.
`IrCutFilterEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = IR cut filter is allowed. `false` = IR cut filter is not allowed.
`BackLightEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Backlight compensation is allowed. `false` = Backlight compensation is not allowed.
`SpeedCtlEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Pan/tilt speed adjustment is allowed. `false` = Pan/tilt speed adjustment is not allowed.
`ProportionalSpeedEnabled`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Proportional speed is allowed. This function is adjusting the movement speed in the image proportional to the zoom level used. `false` = Proportional speed is not allowed.

1. Integer ranges may be overridden by configuration file for driver.
2. Products with Panopsis technology: Cannot be used in Overview mode.

The default parameters set the default value for a specific function in the driver.

[PTZ.Various.V#]

Parameter	Default values	Valid values[1]	Access control	Description
`AutoFocus`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Autofocus is on by default. `false` = Autofocus is off by default.
`AutoIris`	Product/driver dependent	`true` `false`	admin: read, write operator: read viewer: read	`true` = Autoiris is on by default. `false` = Autoiris is off default.

[PTZ.Various.V#] (Continued)

IrCutFilter	Product/driver dependent	on off auto[2]	admin: read, write operator: read viewer: read	on = Default value for IR cut filter is on, that is, the filter will block IR light (daytime use). off = Default value for IR cut filter is off, that is, the filter allows IR light (nighttime use). auto = Default value for IR cut filter is auto, that is, the filter automatically switches between on and off depending on the lighting conditions.
BackLight	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = Backlight compensation is on by default. false = Backlight compensation is off by default.

1. Integer ranges may be overridden by configuration file for driver.
2. Only valid if PTZ.Support.S#.AutoIrCutFilter is true.

The Various parameters have different types of functions.

[PTZ.Various.V#]

Parameter	Default values	Valid values[1]	Access control	Description
CamId	Product/driver dependent	An integer	admin: read, write operator: read viewer: read	Identifies an external PTZ device on a serial port, often set by a dip switch on the external device.
DeviceType	Product/driver dependent	A string	admin: read, write operator: read viewer: read	A driver for an external device may support several variants of the device. Select the matching device type for best compatibility.
Locked	Product/driver dependent	true false	admin: read, write operator: read viewer: read	true = The PTZ position is locked, that is, the position cannot be changed using PTZ commands. false = The PTZ position is unlocked.
LensOffsetX	Product/driver dependent	-9999 ... 9999	admin: read, write operator: read viewer: read	X coordinate of sensor center to lens center vector. Unit is 1/10000 of the sensor width. Used by the areazoom argument. Only applicable if the driver supports lens offset.

[PTZ.Various.V#] (Continued)

LensOffsetY	Product/driver dependent	-9999 ... 9999	admin: read, write operator: read viewer: read	Y coordinate of sensor center to lens center vector. Unit is 1/10000 of the sensor height. Used by the areazoom argument. Only applicable if the driver supports lens offset.
ReturnToOverview	Product/driver dependent	0 ... 300	admin: read, write operator: read viewer: read	Idle timeout; if there is no PTZ activity for this number of seconds, the video channel returns to its home position. The value 0 means that the idle timeout functionality is turned off.
MaxProportionalSpeed	Product/driver dependent	1 ... 1000	admin: read, write operator: read viewer: read	Set the maximum continuous movement speed in 1/100 of fields of view per second. Example: 200 = Max 2.00 fields of view per second.

1. Integer ranges may be overridden by configuration file for driver.

3.3.7 PTZ.UserBasic

This dynamic group is updated when a driver is installed on a video channel. Index # is the video channel number, starting on 1. The parameters in these groups are driver dependent and are hence not known in advance. They are described on the help pages that come with the driver installation.

3.3.8 PTZ.UserAdv

This dynamic group is updated when a driver is installed on a video channel. Index # is the video channel number, starting on 1. The parameters in these groups are driver dependent and are hence not known in advance. Some of the parameters (described below) are supported by most drivers. For the rest of the parameters not described in this section please refer to the help pages that come with the driver installation.

The following parameter is supported by all products where PTZ.Support.SpeedCtl=true.

[PTZ.UserAdv]

Parameter	Default values	Valid values	Access control	Description
MoveSpeed	100	1 ... 100	admin: read, write operator: read viewer: read	Set the default move speed for absolute and relative pan/tilt movements. Can be overridden with the speed argument in the PTZ control HTTP API, see section *3.4.1 PTZ Control*.

The following parameters are supported by most PTZ domes.

[PTZ.UserAdv]

Parameter	Default values	Valid values	Access control	Description

`ImageFreeze`	`off`	`off`
		`on`
		`auto`
			admin: read, write
			operator: read
			viewer: read
				Freeze the image while the Axis product is moving during a pan, tilt or zoom operation
				
				`off` = Image freeze is turned off.
				`on` = Freeze on all movements.
				`auto` = Freeze only when going to presets.

`AutoFlip`	`true`	`true` `false`	admin: read, write operator: read viewer: read	Simulate continuous pan movement in the same direction. See *1.2 Definitions* for further information. `true` = Auto flip is enabled. `false` = Auto flip is disabled.
`MovePrediction`	`false`	`true` `false`	admin: read, write operator: read viewer: read	The Axis product attempts to predict the new position in the pan movement, after compensating for the slight delay while the Axis product changes direction after an auto flip. `true` = Move prediction is enabled. `false` = Move prediction is disabled.

[PTZ.UserAdv] (Continued)

AutoCalibration	true	true false	admin: read, write operator: read viewer: read	Enable auto calibration which automatically resets the components of the camera, pan and tilt if a hardware error occurs. true = Auto calibration is enabled. false = Auto calibration is disabled.
DeviceStatus	cam=ok,pan=ok,tilt=ok	*component=status* [,*component=status...*]	admin: read, write operator: read viewer: read	Shows the status of the camera[1] , pan and tilt hardware components. The values are shown as a comma separated list of *component=status* where statuscan be ok or error[code] where [code] is a 4 digit hexadecimal value. cam = Status for the camera. pan = Status for pan. tilt = Status for tilt.
LastTestDate	A string	A string [date time year]	admin: read, write operator: read viewer: read	Get the date and time for the most recent reset, manually or by AutoCalibration.
DeviceModVer	Product dependent	model:[*model_id*], version:[*version_nbr*]	admin: read, write operator: read viewer: read	Auto detected identifications of internal hardware component model and version. The values are presented as 4 digit hexadecimal values. model = Internal hardware component for model. version = Internal hardware

[PTZ.UserAdv] (Continued)

				component for version.

1. Camera = Camera optics. For example zoom, focus and iris.

3.4 HTTP API

3.4.1 PTZ Control

Control the pan, tilt and zoom behavior of a PTZ unit.

The PTZ control is device-dependent. For information about supported parameters and actual parameter values, check the specification of the Axis PTZ driver used. The following table is only an overview.

Access control: Any user with access to PTZ controls.
Method: `GET/POST`

Syntax:

```
http://<servername>/axis-cgi/com/ptz.cgi?<argument>=<value>[<argument>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
`camera=`*`<int>`*	`1` (default) `...`[1]	Selects the video channel. If omitted the default value `camera=1` is used. This argument is only valid for Axis products with more than one video channel. That is cameras with multiple view areas and video encoders with multiple video channels.
`whoami=`*`<string>`*	*1*	Returns the name of the system-configured device driver.
`center=`*`<int>`*`,`*`<int>`*	`<x>,<y>`	Center the camera on positions x,y where x,y are pixel coordinates in the client video stream.
`areazoom=`*`<int>`*`,`*`<int>`*`,`*`<int>`*	`<x>,<y>,<z>`	Centers on positions `x,y` (like the `center` command) and zooms by a factor of z/100. If z is more than 100 the image is zoomed in (for example; z=300 zooms in to 1/3 of the current field of view). If z is less than 100 the image is zoomed out (for example; z=50 zooms out to twice the current field of view). Note: In some Axis products, the precision of `areazoom` can be strongly improved by calibrating the lens offset parameters.
`imagewidth=`*`<int>`*	`1, ...`[1]	Required in conjunction with `center` or `areazoom` if the image width displayed is different from the default size of the image, which is product-specific.

imageheight=<*int*>	1, ...[1]	Required in conjunction with `center` or `areazoom` if the image height is different from the default size of the image, which is product-specific.
move=<*string*>	home up down left right upleft upright downleft downright stop	**Absolute**:Moves the image 25 % of the image field width in the specified direction. **Relative**: Moves the device approx. 50-90 degrees[2] in the specified direction. `home` = Moves the image to the home position. `up` = Moves the image up. `down` = Moves the image down. `left` = Moves the image to the left. `right` = Moves the image to the right. `upleft` = Moves the image up diagonal to the left. `upright` = Moves the image up diagonal to the right. `downleft` = Moves the image down diagonal to the left. `downright` = Moves the image down diagonal to the right. `stop` = Stops the pan/tilt movement.
pan=<*float*>	-180.0 ... 180.0	Pans the device to the specified absolute coordinates.[3]
tilt=<*float*>	-180.0 ... 180.0	Tilts the device to the specified absolute coordinates.[3]
zoom=<*int*>	1 ... 9999[4]	Zooms the device *n* steps to the specified absolute position. A high value means zoom in, a low value means zoom out.[3]
focus=<*int*>	1 ... 9999	Moves focus *n* steps to the specified absolute position. A high value means focus far, a low value means focus near.
iris=<*int*>	1 ... 9999	Moves iris *n* steps to the specified absolute position. A high value means open iris, a low value means close iris.
brightness=<*int*>	1 ... 9999	Moves brightness *n* steps to the specified absolute position. A high value means brighter image, a low value means darker image.
rpan=<*float*>[5]	-360.0 ... 360.0	Pans the device *n* degrees relative to the current position.[3]
rtilt=<*float*>[5]	-360.0 ... 360.0	Tilts the device *n* degrees relative to the current position.[3]
rzoom=<*int*>[5]	-9999 ... 9999[4]	Zooms the device n steps relative to the current position. Positive values mean zoom in, negative values mean zoom out.
rfocus=<*int*>	-9999 ... 9999	Moves focus *n* steps relative to the current position. Positive values mean focus far, negative values mean focus near.

riris=<*int*>	-9999 ... 9999	Moves iris *n* steps relative to the current position. Positive values mean open iris, negative values mean close iris.
rbrightness=<*int*>	-9999 ... 9999	Moves brightness *n* steps relative to the current position. Positive values mean brighter image, negative values mean darker image.
autofocus=<*string*>	on off	Enable/disable auto focus. on = Enables auto focus. off = Disables auto focus.
autoiris=<*string*>	on off	Enable/disable auto iris. on = Enable auto iris. off = Disable auto iris.
continuouspantiltmove=<*int*>,<*int*>	-100 ... 100,-100 ... 100	Continuous pan/tilt motion. Positive values mean right (pan) and up (tilt), negative values mean left (pan) and down (tilt). 0,0 means stop.[3] Values as <*pan speed*>,<*tilt speed*>.
continuouszoommove=<int>[5]	-100 ... 100	Continuous zoom motion. Positive values mean zoom in and negative values mean zoom out. 0 means stop.
continuousfocusmove=<*int*>	-100 ... 100	Continuous focus motion. Positive values mean focus near and negative values mean focus far. 0 means stop.
continuousirismove=<*int*>	-100 ... 100	Continuous iris motion. Positive values mean iris open and negative values mean iris close. 0 means stop.
continuousbrightnessmove=<*int*>	-100 ... 100	Continuous brightness motion. Positive values mean brighter image and negative values mean darker image. 0 means stop.
auxiliary=<*string*>	<*function name*>	Activates/deactivates auxiliary functions of the device where <*function name*> is the name of the device specific function. Check in driver's documentation or in response to info=1 for information about <*function name*>.
gotoserverpresetname=<*string*>	<*preset name*>[6]	Move to the position associated with the <*preset name*>.
gotoserverpresetno=<*int*>	1, ...[6]	Move to the position associated with the specified preset position number.
gotodevicepreset=<*int*>	<*preset pos*>[6]	Bypasses the presetpos interface and tells the device to go directly to the preset position number <*preset pos*> stored in the device, where the <*preset pos*> is a device-specific preset position number. This may also be a device-specific special function.

speed=<*int*>	1 ... 100	Sets the move speed of pan and tilt.
imagerotation=<*int*>	0 90 180 270	If PTZ command refers to an image stream that is rotated differently than the current image setup, then the image stream rotation must be added to each command with this argument to allow the Axis product to compensate. 0 = Rotate the image 0 degrees. 90 = Rotate the image 90 degrees. 180 = Rotate the image 180 degrees. 270 = Rotate the image 270 degrees.
ircutfilter=<*string*>	auto[1] on off	Control the IR cut filter. auto = Automatically switch between on and off depending on the lighting conditions. on = Apply the filter, that is block IR light. off = Remove the filter, that is allow IR light to reach the image sensor.
backlight=<*string*>	on off	Control the backlight compensation. on = Bright mode. off = Normal mode.
query=<*string*>	limits mode position[2] presetposall presetposcam speed	Returns the current status. limits = PTZ limits for the Axis product. mode = Products with Panopsis technology: The current mode (overview or normal). position = Values for current position. presetposall = Current preset positions for all video channels. presetposcam = Current preset positions for a video channel. speed = Values for pan/tilt speed.
info=<int>	1	Returns a description of available PTZ commands.

1. Product/release dependent. Check the product's release notes.
2. Driver-specific.
3. Products with Panopsis technology: If the product moves to a position where the Panopsis (fisheye) lens is visible, the product will go to Overview mode.
4. If Support.S#.DigitalZoom is true, zoom ranges are 1 ... 19999 for zoom and −19999 ... 19999 for rzoom.
5. Products with Panopsis technology: Does not have any effect in Overview mode.
6. Preset positions are configured using ptzconfig.cgi, see *3.4.2 PTZ Configuration*.

3.4.2 PTZ Configuration

Set and configure PTZ preset positions and On Screen Display (OSD) control.

A server preset saved with name will also get a number and vice versa. You can use both setserverpresetname/removeserverpresetname and setserverpresetno/removeserverpresetno commands on the same preset.

Access control: admin
Method: GET/POST

```
http://<servername>/axis-cgi/com/ptzconfig.cgi?<argument>=<value>[&<argu-
ment>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
camera=*<int>*	1 ...	The video channel. If omitted, the default video channel is used.
osdmenu=*<string>*	open close up down left right select back	Commands to control the OSD menu in the external device. Note that support for different commands, and the behavior of the commands, are driver dependent. open = Open. close = Close. up = Move up. down = Move down. left = Move to the left. right = Move to the right. select = Select. back = Go back.
setserverpresetname=*<string>*	*<preset name>*[1]	Associates the current position to *<preset name>* as a preset position in the Axis product.
setserverpresetno=*<int>*	1 ...	Saves the current position as a preset position number in the Axis product.
home=*<string>*	yes	Makes the current position the home position for the Axis product. Used with setserverpresetname or setserverpresetno.
removeserverpresetname=*<string>*[2]	*<preset name>*[1]	Removes the specified preset position associated with *<preset name>*.
removeserverpresetno=*<int>*[2]	1 ...	Removes the specified preset position.
setdevicepreset=*<int>*	*<preset pos>*	Bypasses the presetpos interface and tells the device to save its current position as preset position *<preset pos>* directly in the device, where *<preset pos>* is a device-specific preset position number. This may also be a device-specific special function

1. <preset name> is a string with a maximum of 31 characters. The following five characters are not allowed: "<>~:
2. When a home preset is removed a new home preset will automatically be created.

3.4.3 Response

3.4.3.1 Success for ptz.cgi

A successful response for ptz.cgi looks like below.

HTTP Code: 204 No content
Content-Type: text/plain

3.4.3.2 Error for ptz.cgi

An error response for `ptz.cgi` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Error:
[message]
```

3.4.3.3 Success for info=1

If PTZ is available a successful response for `info=1` looks like below. The generated response values for this argument depends on what functions the Axis product supports and what functions that are enabled.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Available commands
:
{camera=[n]}
whoami=yes
center=[x],[y]
 imagewidth=[n]
 imageheight=[n]
areazoom=[x],[y],[z]
 imagewidth=[n]
 imageheight=[n]
move={ home | up | down | left | right | upleft | upright | downleft |
downright | stop }
pan=[abspos]
tilt=[abspos]
zoom=[n]
focus=[n]
iris=[n]
brightness=[offset]
rpan=[offset]
rtilt=[offset]
rzoom=[offset]
rfocus=[offset]
riris=[offset]
rbrightness=[offset]
autofocus={ on | off }
autoiris={ on | off }
ircutfilter={ on | off | auto }
backlight={ on | off }
continuouspantiltmove=[x-speed],[y-speed]
continuouszoommove=[speed]
continuousfocusmove=[speed]
continuousirismove=[speed]
continuousbrightnessmove=[speed]
auxiliary=[function]
setserverpresetname=[name]
setserverpresetno=[n]
removeserverpresetname=[name]
removeserverpresetno=[n]
gotoserverpresetname=[name]
gotoserverpresetno=[n]
gotodevicepreset=[n]
speed=[n]
osdmenu=[cmd]
```

```
query={ speed | position | limits | presetposcam | presetposall }
```

3.4.3.4 Success for query=speed

If PTZ is available a successful response for `query=speed` looks like below. The generated response values for this argument depends on what functions the Axis product supports and what functions that are enabled.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
speed=[speed]
```

3.4.3.5 Success for query=position

If PTZ is available a successful response for `query=position` looks like below. The generated response values for this argument depends on what functions the Axis product supports and what functions that are enabled.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
pan=[abspos]
tilt=[abspos]
zoom=[n]
autofocus={ on | off }
autoiris={ on | off }
```

3.4.3.6 Success for query=limits

If PTZ is available a successful response for `query=limits` looks like below. The generated response values for this argument depends on what functions the Axis product supports and what functions that are enabled.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
MinPan=[abspos]
MaxPan=[abspos]
MinTilt=[abspos]
MaxTilt=[abspos]
MinZoom=[n]
MaxZoom=[n]
MinIris=[n]
MaxIris=[n]
MinFocus=[n]
MaxFocus=[n]
MinFieldAngle=[n]
MaxFieldAngle=[n]
MinBrightness=[offset]
MaxBrightness=[offset]
```

3.4.3.7 Success for query=presetposcam

If PTZ is available a successful response for `query=presetposcam` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Preset Positions for camera [n]
presetposno[n]=p[n]
presetposno1=Home
```

3.4.3.8 Success for query=presetposall

If PTZ is available on a multichannel product several cameras can be listed. A successful response for `query=presetposall` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Preset Positions for camera 1
presetposno[n]=p[n]
presetposno1=Home

Preset Positions for camera 2
presetposno[n]=p[n]
presetposno1=Home
...
```

3.4.3.9 Error for query=[invalid value]

An error response for an invalid value, `query=[invalid value]` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Error:
query:  unknown value:  [invalid value]
```

3.4.3.10 Success for whoami=1

If PTZ is available a successful response for `whoami=1` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
[Driver name]
```

3.4.3.11 Success for ptzconfig.cgi

A successful response for `ptzconfig.cgi` looks like below.

HTTP Code: `204 No content`
Content-Type: `text/plain`

3.4.3.12 Error for ptzconfig.cgi

An error response for `ptzconfig.cgi` looks like below.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
Error:
[message]
```

4 PTZ Driver Management API

4.1 Description

Support for different external PTZ devices is realized through drivers that can be uploaded and installed (see page *4.4.1 PTZ Driver Upload* and*4.4.2 PTZ Driver Installation*). This is typically done on Axis products with serial ports. Drivers are installed per video channel. If the product has multiple ports, the port to use for each video channel can be set using the parameter `PTZ.CamPorts.Cam#Port` (see *4.3.1 PTZ.CamPorts*).

For further information about serial ports please refer to the Serial Port API at:
http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php

The PTZ Driver Management API consists of the following CGIs:

`ptzuploader.cgi`	Upload drivers, remove uploaded drivers and list the uploaded drivers.
`ptzupgrader.cgi`	Activate or deactivate a driver for a video channel.

4.2 Prerequisites

4.2.1 Identification

Property: `Properties.API.HTTP.Version=3`
Property: `Properties.PTZ.PTZ=yes`
Property: `Properties.PTZ.DriverManagement=yes`
Firmware: 5.20 and later.
Product category: Axis products with support for uploadable drivers.

4.3 Parameters

4.3.1 PTZ.CamPorts

The `Cam#Port` parameter controls what port is used by PTZ drivers installed on a given video channel `#`.

On products with port managers (`Properties.PTZ.DriverManagement=yes`) the value of the parameter does not directly point to a serial port. Instead it points to a Port Manager. The value of the `Cam#Port` parameter refers to the parameter group, `PortManager.P`[value - 2]. For example, `Cam4Port=2` means that the driver on video channel 4 is using `PortManager.P0`. The Port Manager group contains further port configurations and points to the actual port.

For further information about Port Manager please refer to the Serial Port API at:
http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php

[PTZ.CamPorts]

Parameter	Default values	Valid values	Access control	Description
`Cam#Port`	-1	-1, 1, 2 ...	admin: read, write operator: read viewer: read	Maps video channel to port. The value `-1` means no port used. The index `#` in `Cam#Port` refers to video channel number, starting with `1`.

4.4 HTTP API

Installing a PTZ driver is done in two main steps:

1. Upload the driver (*4.4.1 PTZ Driver Upload*)
2. Install the driver (*4.4.2 PTZ Driver Installation*)

4.4.1 PTZ Driver Upload

This CGI is used for uploading drivers, removing uploaded drivers and listing the uploaded drivers.

Access control: admin
Method: `GET/POST`

Syntax:

```
http://<servername>/axis-cgi/ptz/ptzuploader.cgi?<argument>=<value>[&<argu-
ment>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
`list=`*`<string>`*	`no (default)` `yes`	If `yes`, the Axis product lists the uploaded drivers in the driver database in an XML listing (see below). `no` = The Axis product does not list the drivers. `yes` = The Axis product lists the drivers.
`removedriverid=`*`<int>`*	*<driverid>*	Removes the driver entry identified by *`<driverid>`*from the driver database. The *`<driverid>`* value was returned when the driver was installed, and can also be retrieved using `list`. **Note**: This command only removes the driver entry from the driver database. The driver might still be installed. See *page 31* for information about how to uninstall the driver.

The file content and installation file name, if any, is provided in the HTTP body according to the format given in RFC 1867. See examples below. The body is created automatically by the browser if using an HTML form with input type `file`.

When uploading a new driver, the driver ID is returned in the HTTP response.

```
<?xml version=1.0"?>
<root>
 <id>driverid</id>
</root>
```

The driver listing returned by `list` has the following format.

```
<?xml version=1.0"?>
<root>
 <driverentry>
  <id>driverid</id>
  <driver>drivername</driver>
  <driverversion>driverversion</driverversion>
  <installed><ch>channel1</ch>ch>channel2</ch>...<ch>channeln</ch></installed>
 </driverentry>
 ...
</root>
```

There is one *driverentry* element for each uploaded driver, where:

- *driverid* is the driver ID (returned when uploading a new driver).
- *drivername* is the name of the driver. The parameter `PTZ.PTZDrivers.Driver#` will be set to this value if/when the driver is installed.
- each *<ch>* element contains the number of the video channel where the driver is installed. If there are no *<ch>* elements for a video channel, this could mean that the video channel does not have a driver installed, or that the driver has been removed from the database but not uninstalled.

Example 1:
Upload a driver. The file content of the installation file, `Philips-1.0.ptz` in this example, is provided in the HTTP body as `POST` data.

```
POST /axis-cgi/ptz/ptzuploader.cgi HTTP/1.0
Content-Type:  multipart/form-data; boundary=AaBo3x
Content-Length:  120486

--AaBo3x
Content-Disposition:  form-data; name="Upload Driver"; filename="Philips-1.0.ptz"
Content-Type:  application/octet-stream

<file content of Philips-1.0.ptz>
--AaBo3x--
```

The driver ID, 2 in this example, is returned in the body of the HTTP response.

```
<?xml version="1.0"?>
<root>
 <id>2</id>
</root>
```

Example 2:
List all uploaded drivers.

```
http://myserver/axis-cgi/ptz/ptzuploader.cgi?list=yes
```

If a Pelco driver is installed on channels 3 and 4, the response could be:

```
<?xml version="1.0"?>
<root>
 <driverentry>
  <id>1</id>
  <driver>Pelco</driver>
  <driverversion>2.03</driverversion>
  <installed><ch>3</ch><ch>4</ch></installed>
 <driverentry>
</root>
```

Example 3:
Remove a driver.

```
http://myserver/axis-cgi/ptz/ptzuploader.cgi?removdriverid=5
```

This removes the driver with `ID 5` from the driver database, freeing space on the product. The driver could still be reinstalled and used again.

4.4.2 PTZ Driver Installation

This CGI is used for activating or deactivating a driver for a particular video channel. Drivers may have been previously uploaded, or may be included in the firmware of the product.

At most one driver per channel can be activated using this CGI. In addition, there can be zero or more *static* drivers active for a video channel. The static drivers are built into the product, and cannot be manipulated using this CGI.

When installing a driver, `PortManager.P0.PTZServer.Enabled` and `root.PortManager.P0.SerialCGI.Enabled` will automatically be set to `yes`.

Access control: admin
Method: `GET/POST`

Syntax:

```
http://<servername>/axis-cgi/ptz/ptzupgrader.cgi?<argument>=<value>[<argu-
ment>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
driverid=<int>	`0` (default) . . .	The ID of a driver. This ID has been returned from `ptzuploader.cgi`, or parsed from the XML returned from `ptzuploader.cgi?list=yes`. The value `0` deactivates the currently active driver without installing another driver and sets `PTZ.CamPorts.Cam#Port=-1`
channel=<int>	`1` (default) . . .	Video channel where the driver should be installed. If an uploadable driver already is active for a channel, it will be deactivated when a new driver is installed.

Example 4:
Install driver with ID 2 on default video channel.

```
http://myserver/axis-cgi/ptz/ptzupgrader.cgi?driverid=2
```

Example 5:
Install driver with ID 2 on video channel 1.

```
http://myserver/axis-cgi/ptz/ptzupgrader.cgi?driverid=2&channel=1
```

Example 6:
Uninstall the driver on video channel 1.

```
http://myserver/axis-cgi/ptz/ptzupgrader.cgi?driverid=0&channel=1
```

4.4.3 Response

4.4.3.1 Sucess for ptzuploader.cgi and ptzupgrader.cgi

A successful response for `ptzuploader.cgi` and `ptzupgrader.cgi` looks like below. If `ptzuploader.cgi?list=yes` is requested the response is received in XML as described in *4.4.1 PTZ Driver Upload*.

HTTP Code: `200 OK`
Content-Type: `text/plain`

```
OK
```

5 Queuing API

5.1 Description

The control queue is used to administrate the PTZ control in an environment where several users have PTZ control. A control blocking function lets one user at the time have PTZ control. The other users are put in the control queue. Once a user gained PTZ control PTZ requests can be sent as usual. The rules of the queue are based on what type of access control the user has. For example a user with admin access control will be prioritized over a user with viewer access control. To distinguish individuals using the same user account a cookie is sent the first time the user sends a PTZ request.

5.2 Prerequisites

5.2.1 Identification

Property: `Properties.API.HTTP.Version=3`
Firmware: 5.20 and later.

Products with mechanical PTZ require:

Property: `Properties.PTZ.PTZ=yes`

Products with digital PTZ require:

Property: `Properties.PTZ.DigitalPTZ=yes`

5.3 Parameters

5.3.1 PTZ.Various

[PTZ.Various.V#]

Parameter	Default values	Valid values[1]	Access control	Description
`CtlQueueing`	Product/driver dependent	`true` `false`	admin: read, write operator: read, write viewer: read	The parameter is `true` if control queuing is supported by the driver. If enabled, access to controlling the PTZ unit is limited to the client currently possessing the control. `true` = Control queuing is allowed. `false` = Control queuing is not allowed.

[PTZ.Various.V#] (Continued)

`CtlQueueLimit`	Product/driver dependent	`1 ... 100`	admin: read, write operator: read, write viewer: read	Set the maximal number of clients in a control queue.
`CtlQueuePollTime`	Product/driver dependent	`5 ... 3600`	admin: read, write operator: read, write viewer: read	Set the maximum time in seconds between poll-requests, which an existing client in the control queue must send, to stay active in the queue.

1. Integer ranges may be overridden by configuration file for driver.

5.3.2 PTZ.UserCtlQueue.U#

Parameters for the different users in the control queue. These parameters only have effect if the control queue is enabled (`PTZ.Various.V#.CtlQueuing=true`). In that case, cookies will be required for all calls to `ptz.cgi`.

[PTZ.UserCtlQueue.U#[1]]

Parameter	Default values	Valid values	Access control	Description

`UserGroup`	User group dependent	`Administrator`
		`Operator`
		`Viewer`
		`Event`
		`Autotracking`
		`Guardtour`
		`Recordedtour`
		`Usergroup`
		`...`
			admin: read
			operator: read
				Name of the user group.
				
				`Guardtour` = preset tour (guard tour)
				
				`Recordedtour` = recorded tour (guard tour)
				
				`Autotracking` = mechanical autotracking

`UseCookie`	User group dependent	`yes` `no`	admin: read, write operator: read	If users from a user group shall be separated by using a cookie or by the user name. `yes` = Cookies are used. `no` = Cookies are not used.
`Priority`	User group dependent	`1 ... 100`	admin: read, write operator: read	The priority value. `1` is the highest value.

[PTZ.UserCtlQueue.U#[1]] (Continued)

`TimeoutType`	User group dependent	`timespan`
		`activity`
		`infinity`
			admin: read, write
			operator: read
				Set the timeout type to use.
				
				`timespan` = The user possesses the PTZ control during the time defined by `TimeoutTime`.
				`activity` = The user possesses the PTZ control during the time defined by `TimeoutTime` parameter. The `TimeoutTime` parameter is reset after every PTZ movement.
				`infinity` = The user has infinite PTZ control.

`TimeoutTime`	User group dependent	`1 ... 3600`	admin: read, write operator: read	The time used for each `TimeoutType`. The value is ignored when `TimeoutType` is `infinity`.

1. The # is replaced with a group number starting from `0`, for example PTZ.UserCtlQueue.U5.UserGroup.

5.4 HTTP API

5.4.1 PTZ Control Queue

This CGI handles requests concerning the control queue. If the PTZ control queuing mechanism is enabled (`PTZ.Various.V#.CtlQueuing=true`) for a video channel, control of PTZ units is limited to the client currently possessing it.

Note

Cookies are enabled by default when enabling PTZ control queue.

Access control: viewer with access to PTZ controls
Method: `GET/POST`

Syntax:

```
http://<servername>/axis-cgi/com/ptzqueue.cgi?<argument>=<value>[&<argu-
ment>=<value>...]
```

With the following arguments and values:

Argument	Valid values	Description
control=<*string*>	request drop query	request = Requests PTZ control. drop = Drops the PTZ control or leaves the queue. query = Reports the current status for the client. For possessing clients with no peers existing in the queue, request will reset the control timer. For all other clients, request will have the same effect as query.
camera=<*int*>	1 ...[1]	The video channel number. If omitted, the default channel is used.

1. Product-dependent. Check the product's Release notes.

Example 1:
Request PTZ control for video channel 2.

```
http://myserver/axis-cgi/com/ptzqueue.cgi?control=request&camera=2
```

5.4.2 Response

5.4.2.1 PTZ Control Queue Response

The 200 OK response on success for request and query has a format that enables simple JavaScript parsing.

Success

HTTP Code: 200 OK

Body:

```
<a name="<pos>"</a><a name="<seconds>"</a><a name="<period>"></a>
```

- *<pos>* can have a value from 0 to the maximum value of how many clients are allowed in the queue. This value is the given position in the queue. 0 means that the client is not in the queue, 1 means control is possessed. If the value is 0 the other values are undefined.
- *<seconds>* is the estimated number of seconds remaining, that is for position 1 the remaining control time and for other positions, the time until position 1 is reached. If the value is −1 the time remaining to get control cannot be estimated. This means that a client in the queue has the TimeoutType set to infinity.
- *<period>* is the recommended time in seconds when the client should send a new control=query requests. To stay active in the queue the client must regularly send the request control=query to the Axis product. An inactive client will automatically be removed from the queue.

Example 2:
Control requested.

HTTP Code: 200 OK

```
<a name="3"></a><a name="410"></a><a name="5"></a>
```

This means the client was assigned queue position 3. The expected number of seconds until control is possessed is 410 and the recommended time until the next request is 5 seconds.

Failure

On failure no anchor elements are provided but simply the error message in plain text.

HTTP Code: `200 OK`
Content-Type: text/plain

```
Error:
<error information>
```

VAPIX® version 3
Pan Tilt Zoom API

Ver. M4.2
Date: July 2013
Part No. 52933