

ARCOR
Ihr Internetportal
Empfehlungen von plista 

Heidi Klum und Seal: Noch keine Scheidung [mehr](#)

Dienste [mehr](#)



TOP 2,7% Zinsen + 40€ Prämie für Tagesgeldkonto
Die VTB Direktbank bietet mit 2,7% Zinsen mit die besten Tagesgeldzinsen am Markt.
Zusätzlich gibt...

Axis Camera API, HTTP - Interface Specification

Revision: 1.10

Date: 2002-Nov-13

TABLE OF CONTENTS

- [DOCUMENT HISTORY](#)
- [1 OVERVIEW](#)
 - [1.1 Product and firmware versions](#)
- [2 REFERENCES](#)
- [3 DEFINITIONS](#)
 - [3.1 General notations](#)
 - [3.1.1 General abbreviations](#)
 - [3.1.2 Style convention](#)
 - [3.1.3 General CGI URL syntax and parameters](#)
 - [3.1.4 Parameter value convention](#)
- [4 INTERFACE SPECIFICATION](#)
 - [4.1 Naming conventions and URL syntax](#)
 - [4.1.1 Obsolete CGI parameters](#)
 - [4.2 Server responses](#)
 - [4.2.1 HTTP status codes](#)
- [5 API GROUPS](#)
 - [5.1 General](#)
 - [5.1.1 Get camera parameter values](#)
 - [5.1.2 Camera parameter values response](#)
 - [5.1.3 Set camera parameter values](#)
 - [5.1.4 Get property](#)
 - [5.1.5 Factory default](#)
 - [5.1.6 Restart server](#)
 - [5.1.7 Server report](#)
 - [5.1.8 System logs](#)
 - [5.2 JPEG/MJPEG](#)
 - [5.2.1 JPEG image request](#)
 - [5.2.2 JPEG image \(snapshot\) CGI request](#)
 - [5.2.3 JPEG image response](#)
 - [5.2.4 JPEG buffer request](#)
 - [5.2.5 MJPG video request](#)
 - [5.2.6 MJPG video CGI request](#)
 - [5.2.7 MJPG video response](#)
 - [5.3 MPEG-2](#)
 - [5.3.1 MPEG-2 video request](#)
 - [5.3.2 MPEG-2 video response](#)
 - [5.3.3 MPEG-2 buffer request](#)
 - [5.4 PTZ](#)
 - [5.4.1 PTZ control](#)
 - [5.5 Audio](#)

- 5.5.1 Multipart audio data request
- 5.5.2 Multipart audio data response
- 5.5.3 Multipart audio data transmit
- 5.5.4 Get audio configuration parameters
- 5.6 Motion Detection
 - 5.6.1 Set motion detection parameters
 - 5.6.2 Get motion detection parameters
 - 5.6.3 Get motion detection level
- 5.7 Recording
 - 5.7.1 List recordings
 - 5.7.2 Play recordings
 - 5.7.2.1 Recorded MJPG video request
 - 5.7.2.2 Recorded JPEG image request
 - 5.7.3 Recorded MJPG video response
 - 5.7.4 Preserve and release recordings
 - 5.7.5 Controlling recordings
- 5.8 I/O
 - 5.8.1 I/O control
 - 5.8.1.1 Input
 - 5.8.1.2 Output
 - 5.8.2 Virtual I/O control
 - 5.8.2.1 Input
- 5.9 Serial port
 - 5.9.1 Serial port control
 - 5.9.2 Open serial port
- 5.10 PPP
 - 5.10.1 Close PPP connection
 - 5.10.2 Reset PPP connection maxtimer
- 5.11 Event
 - 5.11.1 Event administration
 - 5.11.1.1 Actions
 - 5.11.1.2 Server responses
- 5.12 Firewall
 - 5.12.1 Firewall administration
 - 5.12.1.1 Actions
 - 5.12.1.2 Parameters
 - 5.12.1.3 Options
 - 5.12.1.4 Server responses

DOCUMENT HISTORY

Version	Date	Comment
1.00	2000-nov-01	Initial version
1.01	2000-nov-13	Added info about dataout and altered description of wait and timeout in "serial port control".
1.02	2000-nov-28	Added entry for controlling image buffers.
1.03	2000-nov-30	Added entries for closing PPP connection and resetting PPP connection.

		PPP connection maxumer.
1.04	2000-dec-11	Added examples for setting motion detection parameters.
1.05	2001-jan-11	Corrected multipart boundary examples.
1.06	2001-jan-16	Removed check of output on IO-ports (until it is supported).
1.07	2001-sep-07	Added Audio support.
1.08	2001-nov-22	Changed audio MIME type and audio parameters.
1.09	2002-apr-15	Added frame rate control.
1.10	2002-nov-13	Added MPEG-2, recording, event and firewall support. Grouped the API requests into different sections.

[TOC](#)

1 OVERVIEW

This document specifies the external HTTP based application programming interface of the Axis camera and video servers.

The HTTP based camera interface provides the functionality to request single and multi-part images, to control camera functions (PTZ, output relay etc.) and to get and set internal parameter values. The image and CGI-requests are handled by the built in Web server of the camera and video servers.

1.1 Product and firmware versions

The support for this stated HTTP API is highly product and release dependent. Please refer to the Release Notes of the actual product for compliance information.

[TOC](#)

2 REFERENCES

HTTP protocol

- [Hypertext Transfer Protocol -- HTTP/1.0](#)

External application programming interfaces (Client side)

- Axis Camera API, HTTP
- Axis Camera Product specific API Notes

[TOC](#)

3 DEFINITIONS

This section contains information of general usage of this document.

3.1 General notations

3.1.1 General abbreviations

The following abbreviations are used throughout this document

CGI *Common Gateway Interface* - a standardized way to communicate between a client (e.g., a web browser) and a server (e.g., a web server).

TBD *To be done/designed* - stated to notify the reader that the referenced section/subsection/entity is intended to be specified but has not reached a level of maturity to be public at this time.

N/A *Not applicable* - a feature/parameter/value is not of any use in a specific task

[TOC](#)

3.1.2 Style convention

In URL syntax and in descriptions of CGI parameters a text in italic within angle brackets denote a content that is to be replaced with either a value or a string. When replacing the text string also the angle brackets shall be replaced. An example of this is the description of the name for the server, denoted with *<servername>* in the URL syntax description below, that is replaced with the string myserver in the URL syntax example, also below.

URL syntax' are written with the "Syntax:" word written in bold face followed by a box with the referred syntax as seen below. The name of the server is written as *<servername>*. This is intended to be replaced with the name of the actual server. This can either be a name, e.g., "thecam" or "thecam.adomain.net" or the associated IP number for the server, e.g., 10.10.2.139.

Syntax:

```
http://<servername>/jpg/image.jpg
```

Description of returned data is written with "Return:" in bold face followed by the returned data in a box. All data returned as HTTP formatted, i.e., starting with the string **HTTP** is line separated with a Carriage Return and Line Feed (CRLF) printed as \r\n.

Return:

```
HTTP/1.0 <HTTP code> <HTTP text>\r\n
```

URL syntax examples are written with "Example:" in bold face followed by a short description and a light grey box with the example.

Example: request default image

```
http://myserver/jpg/image.jpg
```

Examples of what is returned from the server from a request is written with "Example:" in bold face followed by a short description and a light grey box with an example of returned data.

Example: returned data after a successful request

```
HTTP/1.0 200 Ok\r\n
```

[TOC](#)

3.1.3 General CGI URL syntax and parameters

CGI URLs are written in lower-case. CGI parameters are written in lower-case and as one word without any underscores or other separators. When the CGI request includes internal camera parameters, the internal parameters must be written exactly as they are named in the camera or video server. The CGIs are organized in function related directories under the *axis-cgi* directory. The file extension of the CGI is required.

Syntax:

```
http://<servername>/axis-cgi/<subdir>[/<subdir>...]/<cgi>.<ext>
[?<parameter>=<value>[&<parameter>=<value>...]]
```

Example: setting PTZ parameters

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

3.1.4 Parameter value convention

In tables defining CGI parameters and supported parameter values, the default value for optional parameters is system configured.

[TOC](#)

4 INTERFACE SPECIFICATION

4.1 Naming conventions and URL syntax

4.1.1 Obsolete CGI parameters

Some CGI parameters and values in this specification are obsolete and are provided for backward compatibility. They may not be supported in the future.

Obsolete parameters and values are stated in the request descriptions.

4.2 Server responses

4.2.1 HTTP status codes

The built-in Web server uses the standard HTTP status codes.

Return:

```
HTTP/1.0 <HTTP code> <HTTP text>\r\n
```

with the following HTTP code and text meanings

HTTP code	HTTP text	Description
200	OK	The request has succeeded, but an application error can still occur which will be returned as an application error code.

204	No Content	The server has fulfilled the request but there is no new information to send back.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
404	Not Found	The server has not found anything matching the request.

Example: request includes invalid file names.

```
HTTP/1.0 404 Not Found\r\n
```

[TOC](#)

5 API GROUPS

5.1 General

The requests specified in the General section is supported by all camera products.

5.1.1 Get camera parameter values

Get a cameras parameter values.

Note: This request requires administrator access (administrator authorization).

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/admin/getparam.cgi?<parameter>
[&<parameter>...]
```

<parameter>=<value>	Values	Description
<string> ¹	<group>[.<name>]	Returns the value of the camera parameter named <group>.<name> as described in 5.1.2 . If <name> is omitted, all the parameters of the <group> are returned The camera parameters must be entered exactly as they are named in the camera or the video server.

¹ The supported parameters are product/release dependent.

Example: Get the IP address

```
http://myserver/axis-cgi/admin/getparam.cgi?Network.IPAddress
```

Example: Get all of the network parameters

```
http://myserver/axis-cgi/admin/getparam.cgi?Network
```

[TOC](#)

5.1.2 Camera parameter values response

When querying parameter values, the current parameter values are returned.

Successful control requests returns parameter pairs as follows.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<parameter pair>

where <parameter pair> is

<parameter>=<value>\r\n
[ <parameter pair> ]
```

Example: PTZ position query response

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
pan=-50\r\n
tilt=0\r\n
zoom=500\r\n
```

If the CGI request includes an invalid parameter value, the server returns an error message.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Error: <error text (or code)>\r\n
<description>\r\n
```

[TOC](#)

5.1.3 Set camera parameter values

Set a cameras parameter values.

Note: This request requires administrator access (administrator authorization).

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/admin/setparam.cgi?<parameter>=<value>
[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
<string>=<string>	<group.name> = <value>	Assigns <value> to the parameter <group.name> The <value> must be URL encoded when it contains non-alphanumeric characters.

		The camera parameters must be entered exactly as they are named in the camera or the video server.
nosync=<string> ¹	yes	Specifies that there should be no sync (write) of the corresponding configuration file on flash. If parameter is omitted, a sync will occur.

Example: Set default image resolution to 320x240 pixels

```
http://myserver/axis-cgi/admin/setparam.cgi?Image.Resolution=320x240
```

Example: Set the bandwidth limitation to 500, without writing the corresponding configuration file on flash

```
http://myserver/axis-cgi/admin/setparam.cgi?Network.Bandwidth=500&nosync=yes
```

[TOC](#)

5.1.4 Get property

Get property

Note: This request requires administrator access (administrator authorization).

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/admin/getproperty.cgi?[<parameter>[&<parameter>...]]
```

with the following parameter and values

<parameter>=<value>	Values	Description
<string> ¹	<name>	Returns the value of the camera property named <name> as described in 5.1.2 . If <name> is omitted, all the available properties are returned.

¹ The supported parameters are product/release dependent.

Example: Get the product name

```
http://myserver/axis-cgi/admin/getproperty.cgi?productname
```

[TOC](#)

5.1.5 Factory default

Reload factory default

Note: This request requires administrator access (administrator authorization).

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/factorydefault.cgi
```

[TOC](#)

5.1.6 Restart server

Restart server

Note: This request requires administrator access (administrator authorization).

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/restart.cgi
```

[TOC](#)

5.1.7 Server report

This CGI request generates and returns a server report. This report is useful as an input to the support organization when support is required. The report includes product information, parameter settings and system logs.

Note: This request requires administrator access (administrator authorization).

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/serverreport.cgi
```

[TOC](#)

5.1.8 System logs

Get system log information

Note: This request requires administrator access (administrator authorization).

Note: The response is product/release dependent.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/systemlog.cgi
```

Return:

```
HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\n<system log information>
```

[TOC](#)

5.2 JPEG/MJPEG

The requests specified in the JPEG/MJPEG section is supported by those camera products that uses JPEG/MJPEG encoding.

5.2.1 JPEG image request

Request JPEG image

Method: GET

Syntax:

```
http://<servername>/jpg[<camera>]/<name>.jpg
```

with the following parameters

Parameter	Values	Description
<camera>	1, ... ¹	Select input source. Applies only to servers with more than one source of input. <i>Default:</i> default camera
<name>	image, quad ¹ , halfsize ² , fullsize ² , hugesize ²	"image" returns an image with the default resolution and compression as defined in the system configuration. The <camera> option is not allowed in quad request.

¹ Product dependent. Check product specification.

² Obsolete.

Example: request JPEG image from default camera with default resolution and compression

```
http://myserver/jpg/image.jpg
```

[TOC](#)

5.2.2 JPEG image (snapshot) CGI request

Request a JPEG image (snapshot) with specified properties.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/jpg/image.cgi  
[?<parameter>=<value>[&<parameter>=<value>...]]
```

<parameter>=<value>	Values	Description
resolution=<int>x<int>	<width> ¹ ,<height> ¹	Specify the resolution as <width> times <height> number of pixels of the returned image.
camera=<int>	1, ... ¹	Applies only to video servers with more than one video input. Selects the source camera.
compression=<int>	0 - 100 ¹	Adjusts the compression level of the image. Higher value corresponds to higher compression, i.e. lower quality and smaller image size.

		Note: This value is internally mapped and is therefore product dependent.
colorlevel=<int>	0 - 100 ¹	Sets level of color or grayscale. 0 = grayscale, 100 = full color. Note: This value is internally mapped and is therefore product dependent.
clock=<int>	0, 1	Shows/hides the time stamp. 0 = hide, 1 = show
date=<int>	0, 1	Shows/hides the date. 0 = hide, 1 = show
quad=<int>	0, 1 ¹	Generate a quad image. 0 = normal, 1 = quad
text=<int>	0, 1	Shows/hides the text. 0 = hide, 1 = show
rotation=<int>	0, 90, 180, 270 ¹	Rotates the image clockwise.
showlength=<int>	0, 1	Content-Length is added to the HTTP-header and in the boundary section, between the images. 0 = hidden, 1 = shown.

¹ Product dependent. Check product specification.

Example: a JPEG image from camera 1 with a resolution of 320x240 and compression of 25

```
http://myserver/axis-cgi/jpg/image.cgi?resolution=320x240&camera=1
&compression=25
```

TOC

5.2.3 JPEG image response

When a jpeg image is requested, the server either returns the specified JPEG image file or with an error.

An optional field "Content-Length" header entry specifying the image size in bytes <image size> may also be included if the camera or video server is configured to include it. The also optional field "Content-Auth" is followed by authorization specific data <authorization information>, e.g., which encryption method that is used.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
[ Content-Length: <image size>\r\n ]
[ Content-Auth: <authorization information>\r\n ]
\r\n
```

```
<JPEG image data>\r\n
```

Example: requested JPEG image

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15656\r\n
\r\n
<JPEG image data>\r\n
```

TOC

5.2.4 JPEG buffer request

Request for controlling image buffers through HTTP.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/buffer/command.cgi?<parameter>=<value>
[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
do=<string>	start, stop, get, reset	"start" will create a new image buffer. "stop" will make the image buffer store post alarm images. Buffer will stop after all post alarm images is taken. "get" is used to fetch an image in the image buffer. "reset" will restart the image buffer, removing all previous images.
buffername=<string>	<any string>	Name used for identifying the buffer.
uri=<string>	<any string>	Corresponding image URI to be used by the image buffer. Note: Must be URI encoded. The URI should also begin with "ftp://", otherwise an HTTP header is added to the image.
prealarm=<int>	0, ...	Number of images to be stored in the pre alarm buffer.
postalarm=<int>	0, ...	Number of images to be saved after an alarm occurred.
delay=<int>	<milliseconds>	The preferred time between images in milliseconds.
index=<int>	<image number>	The index of image in buffer.

Example 1: Create an image buffer, named DOOR1, with 10 pre alarm image and 15 post alarm images.

```
http://myserver/axis-cgi/buffer/command.cgi?do=start&buffername=DOOR1
```

```
&prealarm=10&postalarm=15&delay=1&uri=ftp://jpg/1/image.jpg
```

Example 2: Stop a buffer.

```
http://myserver/axis-cgi/buffer/command.cgi?do=stop&buffername=DOOR1
```

Example 3: Get images from a buffer.

```
http://myserver/axis-cgi/buffer/command.cgi?do=get&buffername=DOOR1&index=1
```

[TOC](#)

5.2.5 MJPG video request

Request Multipart JPEG image

Method: GET

Syntax:

```
http://<servername>/mjpg[/<camera>]<name>.mjpg
```

with the following parameters

Parameter	Values	Description
<camera>	1, ... ¹	Select input source. Applies only to servers with more than one source of input. <i>Default:</i> default camera
<name>	video, quad ¹ , halfsize ² , fullsize ² , hugesize ²	"video" returns a multipart image stream with the default resolution and compression as defined in the system configuration. The <camera> option is not allowed in quad request.

¹ Product dependent. Check product specification.

² Obsolete.

Example: request JPEG image stream from the 2nd camera with default resolution and compression

```
http://myserver/mjpg/2/video.mjpg
```

[TOC](#)

5.2.6 MJPG video CGI request

Request a Multipart JPEG image stream (video) with specified properties.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/mjpg/video.cgi  
[?<parameter>=<value> [&<parameter>=<value>...]]
```

<parameter>=<value>	Values	Description
resolution=<int>x<int>	<width> ¹ ,<height> ¹	Specify the resolution as <width> times <height> number of pixels of the returned image.
camera=<int>	1, ... ¹	Applies only to video servers with more than one video input. Selects the source camera. If omitted, the default camera is used.
compression=<int>	0 - 100 ¹	Adjusts the compression level of the image. Higher value corresponds to higher compression, i.e. lower quality and smaller image size. Note: This value is internally mapped and is therefore product dependent.
colorlevel=<int>	0 - 100 ¹	Sets level of color or grayscale. 0 = grayscale, 100 = full color. Note: This value is internally mapped and is therefore product dependent.
clock=<int>	0, 1	Shows/hides the time stamp. 0 = hide, 1 = show
date=<int>	0, 1	Shows/hides the date. 0 = hide, 1 = show
quad=<int>	0, 1 ¹	Generate a quad image. 0 = normal, 1 = quad
text=<int>	0, 1	Shows/hides the text. 0 = hide, 1 = show
rotation=<int>	0, 90, 180, 270 ¹	Rotates the image clockwise.
showlength=<int>	0, 1	Content-Length is added to the HTTP-header and in the boundary section, between the images. 0 = hidden, 1 = shown.
duration=<int>	0, ...	Specifies for how many seconds the video will be generated and pushed to the client.
nbrofframes=<int>	1, ...	Specifies how many frames the server shall generate and push.
		Using either req_fps or des_fps (these can not be used at the same time) it is possible to specify the frame rate from the server. <small>req_fps = required FPS</small>

req_fps des_fps=<int>	1, ...	<i>req_fps = requested FPS</i> Required FPS has higher priority than desired FPS if these are used in two different requests simultaneously. <i>Use req_fps for streams with high priority and des_fps for less important streams.</i>
deltatime=<int>	0, 1	Timediff, specifying the time between the images in us, is added to the boundary section between the images. 0 = hidden, 1 = shown.

¹ Product dependent. Check product specification.

Example: a Multipart JPEG image stream from camera 1 with a resolution of 320x240 and compression of 25

```
http://myserver/axis-cgi/mjpg/video.cgi?resolution=320x240&camera=1
&compression=25
```

Example: a Multipart JPEG image stream from camera 1 with a required frame rate of 5

```
http://myserver/axis-cgi/mjpg/video.cgi?req_fps=5
```

TOC

5.2.7 MJPG video response

When a MJPG video is requested, the server returns a continuously flow of jpeg files. The content type is "multipart/x-mixed-replace" and each image ends with a boundary string <boundary>. The returned image and HTTP data is equal to the request for a single JPEG image.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<image>
```

where the proposed <boundary> is

myboundary

and the returned <image> field is

```
Content-Type: image/jpeg\r\n
[ Content-Length: <image size>\r\n ]
[ Content-Auth: <authorization information>\r\n ]
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
<image>
```

Example: requested Multipart JPEG image

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
myboundary\r\n
```

```
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15656\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 14978\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15136\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
.
.
```

[TOC](#)

5.3 MPEG-2

The requests specified in the MPEG-2 section is supported by those camera products that uses MPEG-2 encoding.

5.3.1 MPEG-2 video request

Request an MPEG-2 video stream with specified properties.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/mpeg2/video.cgi[?<parameter>=<value>[&<parameter>=<value> ...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
duration=<int>	0, ...	Specifies for how many seconds the video shall be generated and pushed to the client. If the duration parameter is left out, an infinite stream of video will be provided to the client.
camera=<int>	1, ... ¹	Searched recording should match this camera id
maxlag=<int>	500,... ¹	A number specifying at which lag in ms the server should eliminate the lag by cutting the stream. A lag could occur at the client due to clock differences between the client computer and the server. Note: This will be the maximum lag. The real lag could be lower than the specified value. Values lower than 500 will be interpreted as 500. If the maxlag parameter is left out, the maximum lag will be determined by the available space in the server.

¹ Product dependent. Check product specification.

[TOC](#)

5.3.2 MPEG-2 video response

When an MPEG-2 video is requested, the server either returns the specified video stream or an error message.

A field "X-Prebuffer-Length" is included in the HTTP header when responding to an MPEG-2 buffer request. This entry specifies the length in seconds of the pre buffer video that is included in the video stream response. This length depends on what is specified in the request and a maximum value that is set by the server. If the requested length is greater than or equal to the maximum value set by the server, the returned pre buffer length will be equal to the maximum value. If the requested length is shorter than the maximum value, the returned length will be equal to the requested length.

```
HTTP/1.0 200 OK \r\n
Content-type: video/mpeg\r\n
[X-Prebuffer-Length: <floating number in seconds> \r\n]
\r\n
<MPEG stream>\r\n
```

Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: video/mpeg\r\n
\r\n
<MPEG stream>\r\n
```

[TOC](#)

5.3.3 MPEG-2 buffer request

Request for pre buffered MPEG-2 video.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/mpeg2(buffer.cgi?[<parameter>=<value>]&<parameter>=<value> ...])
```

with the following parameters and values

<parameter>=<value>	Values	Description
prebuffer=<int>	0, ...	Number of seconds of video, buffered before the request, to be provided to the client.
duration=<int>	0, ...	Specifies for how many seconds the video shall be generated and pushed to the client. If the duration parameter is left out, an infinite stream of video will be provided to the client.

[TOC](#)

5.4 PTZ

The requests specified in the PTZ section is supported by those camera products that have support for Pan/Tilt/Zoom devices.

5.4.1 PTZ control

To control the Pan, Tilt, and Zoom behavior of a PTZ unit, the following PTZ control URL is used.

Important: The PTZ control is device dependent. Please study the specification of the Axis PTZ driver you are going to use to get information about supported parameters and the actual parameter values. The following table is just an overview.

Method: GET/POST

Syntax:

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

with the following meanings

<parameter>=<value>	Values	Description
camera=<int>	1, ... ¹	Applies only to video servers with more than one video input. Selects the source camera. If omitted, the default camera is used.
whoami=<string>	<any value>	Returns the name of the system configured device driver.
center=<int>,<int> extrem ³ =<int>,<int>	<x>,<y>	<p>Absolute: Used to send the coordinates for the point in the image where the user clicked. This information is then used by the server to calculate the pan/tilt move that is required to center (approximately) the clicked point.</p> <p>Relative: Used to send the coordinates for the point in the image where the user clicked. This information is then used by the server to calculate direction and number of degrees to move. The number of degrees increases with the distance from the center of the image to the point clicked.</p>
imagewidth=<int>	1, ... ¹	Needed in conjunction with <i>center</i> if the image width is displayed different from the default size of the image which is product specific.
imageheight=<int>	1, ... ¹	Needed in conjunction with <i>center</i> if the image height is different from the default size which is product specific.
move=<string>	home, up, down,	Absolute: Moves the device 5 degrees in the specified direction.

	left, right, upleft, upright, downleft, downright	Relative: Moves the device approx. 50-90 degrees ² in the specified direction.
pan=<float>	-180.0 - 180.0	Absolute: Pans the device relative to the (0,0) position. Relative: N/A
tilt=<float>	-180.0 - 180.0	Absolute: Tilts the device relative to the (0,0) position. Relative: n/a
zoom=<int>	1 - 9999	Absolute: Zooms the device <i>n</i> steps relative to the home position. Relative: n/a
focus=<int>	1 - 9999	Absolute: Move Focus <i>n</i> steps relative to the home position. Relative: n/a
iris=<int>	1 - 9999	Absolute: Move iris to <i>n</i> steps relative to the home position. Relative: n/a
rpan=<float>	-360.0 - 360.0	Absolute: Pans the device <i>n</i> degrees relative to the current position. Relative: Pans the device approx. <i>n</i> degrees relative to the current position.
rtilt=<float>	-360.0 - 360.0	Absolute: Tilts the device <i>n</i> degrees relative to the current position. Relative: Tilts the device approx. <i>n</i> degrees relative to the current position.
rzoom=<int>	-9999 - 9999	Absolute: Zooms the device <i>n</i> steps relative to the current position. Positive values means zoom in, negative values means zoom out. Relative: Zooms the device approx. <i>n</i> steps relative to the current position. Positive values means zoom in, negative values means zoom out.
rfocus=<int>	-9999 - 9999	Absolute: Move Focus <i>n</i> steps relative to the current position. Positive values means focus near, negative values means focus far. Relative: Move Focus approx. <i>n</i> steps relative to the current position. Positive values means focus near,

		negative values means focus far.
riris=<int>	-9999 - 9999	Absolute: Move iris <i>n</i> steps relative to the current position. Positive values means open iris, negative values means close iris. Relative: Move iris approx. <i>n</i> steps relative to the current position. Positive values means open iris, negative values means close iris.
zoomrel3=<string>	tele, wide, telemax, telemin	Adjusts the zoom gradually.
focusrel3=<string>	far, farmore, near, nearmore	Adjusts the focus gradually.
irisrel3=<string>	open, close, openmore, closemore	Adjusts the iris gradually.
autofocus=<string>	on, off	Autofocus On/Off.
autoiris=<string>	on, off	Autoiris On/Off.
continuouspantiltmove=<int>,<int>	-100 - 100, -100 - 100	Continuous pan/tilt motion. Positive values means right (pan) and up (tilt), negative values means left (pan) and down (tilt). "0,0" means stop. Values as < <i>pan speed</i> >,< <i>tilt speed</i> >
continuouszoommove=<int>	-100 - 100	Continuous zoom motion. Positive values means zoom in and negative values means zoom out. "0" means stop.
continuousfocusmove=<int>	-100 - 100	Continuous focus motion. Positive values means focus near and negative values means focus far. "0" means stop.
continuousirismove=<int>	-100 - 100	Continuous iris motion. Positive values means iris open and negative values means iris close. "0" means stop.
auxiliary=<string>	< <i>function name</i> >	Activates/deactivates auxiliary functions of the device where < <i>function name</i> > is the name of the device specific function.
setserverpresetname=<string>	< <i>preset name</i> > ⁴	Associates the current position to < <i>preset name</i> > as a preset position in the server.
setserverpresetno=<int>	1, ...	Saves the current position as a preset position number in the server.
removeserverpresetname=<string>	< <i>preset name</i> > ⁴	Removes the specified preset position associated with < <i>preset name</i> >

<i><summary></i>		<i><preset name></i>
removeserverpresetno=<int>	1, ...	Removes the specified preset position.
gotoserverpresetname=<string>	<i><preset name></i> ⁴	Move to the position associated with the <i><preset name></i> .
gotoserverpresetno=<int>	1, ...	Move to the position associated with the specified preset position number.
setdevicepreset=<int>	<i><preset pos></i>	Bypasses the presetpos interface and tells the device to save it's current position as preset position <i><preset pos></i> directly in the device, where <i><preset pos></i> is a device specific preset position number.
gotodevicepreset=<int>	<i><preset pos></i>	Bypasses the presetpos interface and tells the device to go directly to the preset position number <i><preset pos></i> that is stored in the device, where the <i><preset pos></i> is a device specific preset position number.
barcoord=<int>,<int>	<i><x>,<y></i>	Used in conjunction with panbar, tiltbar, zoombar, focusbar or irisbar to send coordinates to the server.
panbar=<int>,<string>	<i><length>,<alignment></i>	<p><i><length></i> is the length of the bar in pixels which is needed in order to calculate the middle of the bar.</p> <p><i><alignment></i> is one of the strings "horisontal" or "vertical".</p> <p>The alignment string determines if the x (horizontal) or the y (vertical) coordinate from <i>barcoord</i> is used i.e. if the bar is horizontal use "horizontal" and if the bar is vertical use "vertical" as alignment.</p>
tiltbar=<int>,<string>	<i><length>,<alignment></i>	<p><i><length></i> is the length of the bar in pixels which is needed in order to calculate the middle of the bar.</p> <p><i><alignment></i> is one of the strings "horisontal" or "vertical".</p> <p>The alignment string determines if the x (horizontal) or the y (vertical) coordinate from <i>barcoord</i> is used i.e. if the bar is horizontal use "horizontal" and if the bar is vertical use "vertical" as alignment.</p>
zoombar=<int>,<string>	<i><length>,<alignment></i>	<p><i><length></i> is the length of the bar in pixels which is needed in order to calculate the middle of the bar.</p> <p><i><alignment></i> is one of the strings "horisontal" or "vertical".</p> <p>The alignment string determines if the x (horizontal) or the y (vertical) coordinate from <i>barcoord</i> is used i.e. if the bar is horizontal use "horizontal" and if the bar is vertical use "vertical" as alignment.</p>

vertical as alignment		
focusbar=<int>,<string>	<length>, <alignment>	<p><length> is the length of the bar in pixels which is needed in order to calculate the middle of the bar.</p> <p><alignment> is one of the strings "horizontal" or "vertical".</p> <p>The alignment string determines if the x (horizontal) or the y (vertical) coordinate from <i>barcoord</i> is used i.e. if the bar is horizontal use "horizontal" and if the bar is vertical use "vertical" as alignment.</p>
irisbar=<int>,<string>	<length>, <alignment>	<p><length> is the length of the bar in pixels which is needed in order to calculate the middle of the bar.</p> <p><alignment> is one of the strings "horizontal" or "vertical".</p> <p>The alignment string determines if the x (horizontal) or the y (vertical) coordinate from <i>barcoord</i> is used i.e. if the bar is horizontal use "horizontal" and if the bar is vertical use "vertical" as alignment.</p>
speed=<int>	1 - 100	Sets the head speed of the device connected to the specified camera.
query=<string>	speed, position, presetposcam, presetposall	Returns the current parameter values. See 5.1.2 for more information about server responses.
info=<int>	1	<p>Returns a description of this CGI-request.</p> <p>No PTZ control is performed.</p>

¹ Product dependent. Check product specification.

² The actual values are device driver specific.

³ Obsolete.

⁴ <preset name> is a string with a maximum of 31 characters, ~ is not allowed.

TOC

5.5 Audio

The requests specified in the Audio section is supported by those camera products that have audio capability.

5.5.1 Multipart audio data request

Request a Multipart Audio stream.

Method: GET

Syntax:

```
| http://<servername>/axis-cgi/audio/receive.cgi
```

there are no valid parameters and values

Example: a Multipart Audio stream

```
http://myserver/axis-cgi/audio/receive.cgi
```

[TOC](#)

5.5.2 Multipart audio data response

When an audio stream is requested/transmitted, the server returns/receives a continuous flow of audio packets. The content type is "multipart/x-mixed-replace" and each audio packet ends with a boundary string <boundary>. The MIME type used for the audio transmitted is audio/32KADPCM. The message body contains a block of binary data.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<audio>
```

where the proposed <boundary> is

myboundary

and the <audio> field is

```
Content-Type: audio/32KADPCM\r\n
<Audio data>\r\n
--<boundary>\r\n
<audio>
```

Example: Multipart Audio data

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
.
.
.
```

[TOC](#)

5.5.3 Multipart audio data transmit

transmit a Multipart Audio data stream.

Method: POST

Syntax:

```
http://<servername>/axis-cgi/audio/transmit.cgi
```

there are no valid parameters and values

Example: transmit a Multipart Audio stream

```
http://myserver/axis-cgi/audio/transmit.cgi
```

[TOC](#)

5.5.4 Get audio configuration parameters

Get information about audio configuration parameters.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/audio/getparam.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<audio parameters>
```

and <audio parameters> is

```
root.Audioc.duplexMode=<duplexMode>\r\n
root.Audioc.maxNumberOfClients=<maxNumberOfClients>\r\n
root.Audioc.forwardOnPost=<forwardOnPost>\r\n
root.Audioc.buttonMode=<buttonMode>\r\n
root.Audioc.clientBufferSize=<clientBufferSize>\r\n
root.Audioc.connectedClients=<connectedClients>\r\n
root.Audioc.connectedPostClients=<connectedPostClients>\r\n
\r\n
```

Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
root.Audioc.duplexMode=half\r\n
root.Audioc.maxNumberOfClients=1\r\n
root.Audioc.forwardOnPost=yes\r\n
root.Audioc.buttonMode=push\r\n
root.Audioc.clientBufferSize=400\r\n
root.Audioc.connectedClients=2\r\n
root.Audioc.connectedPostClients=1\r\n
\r\n
```

[TOC](#)

5.6 Motion Detection

The requests specified in the Motion Detection section is supported by those camera products that have built-in motion detection.

5.6.1 Set motion detection parameters

Adding new motion detection window, removing or updating any existing window.

The "key" entry is a unique window identity communicated in all actions concerning the window, i.e., when adding, updating or removing it, or when reading information about it. The "name" entry is a window identification useful e.g., when communicating with a user in an application or an alarm.

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/admin/mdsetparam.cgi?<parameter>=<value>
[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
action=<string>	add, remove, update	<p>"add" creates a new motion detection window and requires both the "key" and "name" entries to be defined.</p> <p>"update" make changes to an existing window and requires only the "key" entry to be defined.</p> <p>"remove" deletes an existing window and requires only the "key" entry to be defined. Any other additional entry is ignored.</p> <p><i>Default:</i> no default value provided.</p>
key=<int>	0, ... ¹	<p>A unique identity to the motion detection window.</p> <p><i>Default:</i> no default value provided.</p>
name=<string>	<any string>	<p>A user's window identification, e.g., "Door" or "Window".</p> <p><i>Default:</i> no default value provided.</p>
method=<char>	w	<p>Motion detection method to use for this window.</p> <p><i>Default:</i> w</p>
left=<int>	0 ... 9999	<p>The coordinate for the left boundary of the rectangular motion detection window.</p> <p>The full value range is related to the full image width and 0 is counted from the left hand side of the image.</p> <p><i>Default:</i> 0</p>
right=<int>	0 ... 9999	The coordinate for the right boundary of the rectangular

		<p>motion detection window.</p> <p>The full value range is related to the full image width and 0 is counted from the left hand side of the image.</p> <p><i>Default:</i> 9999</p>
top=<int>	0 ... 9999	<p>The coordinate for the upper boundary of the rectangular motion detection window.</p> <p>The full value range is related to the full image height and 0 is counted from the upper side of the image.</p> <p><i>Default:</i> 0</p>
bottom=<int>	0 ... 9999	<p>The coordinate for the bottom boundary of the rectangular motion detection window.</p> <p>The full value range is related to the full image height and 0 is counted from the upper side of the image.</p> <p><i>Default:</i> 9999</p>
sensitivity=<int>	0 ... 100	<p>Image intensity sensitivity.</p> <p><i>Default:</i> method specific</p>
history=<int>	0 ... 100	<p>Time dependency for reference image update.</p> <p><i>Default:</i> method specific</p>
size=<int>	0 ... 100	<p>Object size description.</p> <p><i>Default:</i> method specific</p>

Return: all parameters are set correct

```
HTTP/1.0 204 No Content\r\n
```

Return: one or more critical errors occurred

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Error: <error description>\r\n
```

Example: Add a new motion detection window

```
http://myserver/axis-cgi/admin/mdsetparam.cgi?action=add&key=2
&name=Entrance&top=500&bottom=7000&left=5000&right=8500
```

Example: Update parameters for an existing motion detection window

```
http://myserver/axis-cgi/admin/mdsetparam.cgi?action=update&key=2
&top=1500&bottom=8000
```

Example: Remove a motion detection window

```
http://myserver/axis-cgi/admin/mdsetparam.cgi?action=remove&key=2
```

[TOC](#)

5.6.2 Get motion detection parameters

Read information about current defined motion detection windows.

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/motion/mdgetparam.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<window data>
```

and <window data> is

```
key=<key>\r\n
name=<name>\r\n
method=<method>\r\n
left=<left>\r\n
top=<top>\r\n
right=<right>\r\n
bottom=<bottom>\r\n
sensitivity=<sensitivity>\r\n
history=<history>\r\n
size=<size>\r\n
\r\n
[ <window data> ]
```

Example: two motion detection windows "Backdoor" and "Window"

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
key=0\r\n
name=Backdoor\r\n
method=w\r\n
left=200\r\n
top=800\r\n
right=3600\r\n
bottom=9400\r\n
sensitivity=90\r\n
history=85\r\n
size=10\r\n
\r\n
key=1\r\n
name=Window\r\n
method=w\r\n
left=7000\r\n
top=1000\r\n
right=9500\r\n
bottom=4000\r\n
sensitivity=95\r\n
history=85\r\n
size=25\r\n
\r\n
```

[TOC](#)

5.6.3 Get motion detection level

To read the current motion detection levels of all windows, the URL stated below is used.

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/motion/motiondetect.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<motion levels>
```

where the proposed boundary *<boundary>* is

axismdb

and the *<motion levels>* part is

```
Content-Type: text/plain\r\n
\r\n
<motion level for window 0>
--<boundary>\r\n
```

and *<motion level for window n>*" is

```
key=<key n>;level=<motion level for n>;
threshold=<threshold level for n>;\r\n
[ <motion level for window n+1> ]
```

Example: two motion detection windows with key "0" and key "1"

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=axismdb\r\n
\r\n
--axismdb\r\n
Content-Type: text/plain\r\n
\r\n
key=0;level=28;threshold=45;\r\n
key=1;level=43;threshold=25;\r\n
--axismdb\r\n
Content-Type: text/plain\r\n
\r\n
key=0;level=54;threshold=45;\r\n
key=1;level=38;threshold=25;\r\n
--axismdb\r\n
Content-Type: text/plain\r\n
\r\n
key=0;level=49;threshold=45;\r\n
key=1;level=19;threshold=25;\r\n
--axismdb\r\n
.
.
.
```

[TOC](#)

5.7 Recording

The requests specified in the Recording section is supported by those camera products that provides storage of recordings.

5.7.1 List recordings

Choose to list all recordings, one specific recording or all recordings matching a criteria.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/view/reclist.cgi[?<parameter>=<value>
[&<parameter>=<value> ...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
recordingid=<int>	> 0	recording id, unique number for every recording. Only this recording will be listed
camera=<int>	1, ... ¹	Searched recording should match this camera id
event=<int>	> 0	Searched recording should match this event id. Id of a event is got from the event parameter.
time=<int>	Seconds since 1970: 1/100 seconds	Searched recording should match this time.

¹ Product dependent. Check product specification.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<recording data>

and <recording data> is

recordingid=<recordingid>\r\n
cameraid=<cameraid>\r\n
cameraname=<name>\r\n
eventid=<eventid>\r\n
eventpath=<eventpath>\r\n
eventname=<name>\r\n
prebufferstartframe=<prebufferstartframe>\r\n
prebufferstarttime=<prebufferstarttime>\r\n
eventstartframe=<eventstartframe>\r\n
```

```

eventstarttime=<eventstarttime>\r\n
eventstopframe=<eventstopframe>\r\n
eventstoptime=<eventstoptime>\r\n
postbufferstopframe=<postbufferstopframe>\r\n
postbufferstoptime=<postbufferstoptime>\r\n
priority=<priorty>\r\n
ongoing=<ongoing>\r\n
secured=<secured>\r\n
startquality=<startquality>\r\n
stopquality=<stopquality>\r\n
\r\n
[ <recording data> ]

```

with the following parameters and values

<parameter>=<value>	Values	Description
recordingid=<int>	> 1	recording id, unique number for every recording
camera=<int>	1, ... ¹	camera id
cameraname=<string>	user specified	Name parameter of the camera
eventid=<int>	user specified	event id
eventpath=<string>	root.Event.E<id>	Path to event in parameter handler
eventname=<string>	user specified	Name parameter of the event
prebufferstartframe=<int>:<int>	id of startframe	Startframe is the first image/frame of the recording. If the beginning is deleted by reducing the startframe is moved to the new start. 3765734:98
prebufferstarttime=<int>:<int>	Seconds since 1970: 1/100 seconds	Time of the startframe 983453412:10
eventstartframe=<int>:<int>	frame id	The image/frame when the event was triggered 3765734:98
eventstarttime=<int>:<int>	Seconds since 1970: 1/100 seconds	Time of the eventstartframe
eventstopframe=<int>:<int>	frame id	The image/frame when the event was ended 3765734:98
eventstoptime=<int>:<int>	Seconds since 1970: 1/100 seconds	Time of the eventstopframe

postbufferstopframe=<int>: <int>	frame id	The last image/frame of recording 3765734:98
postbufferstoptime=<int>: <int>	Seconds since 1970: 1/100 seconds	Time of the postbufferstopframe
priority=<int>	0 - 100 ¹	Priority of the event 100 = highest priority Note: This value is internally mapped and is therefore product dependent.
ongoing=<string>	yes, no	Still recording or not.
secured=<string>	yes, no	A preserved recording will not be reduced or deleted.
startquality=<int>	0 - 100 ¹	Quality of the beginning of the recording, due to reduction. 100 = highest priority Note: This value is internally mapped and is therefore product dependent.
stopquality=<int>	0 - 100 ¹	Quality of the beginning of the recording, due to reduction. 100 = highest priority Note: This value is internally mapped and is therefore product dependent.

¹ Product dependent. Check product specification.

Example 1: List recording id 12

Request:

<http://myserver/axis-cgi/view/reclist.cgi?recordingid=12>

Response:

HTTP/1.0 200 OK\r\n

Content-Type: text/plain\r\n

\r\n

recordingid=12\r\n

cameraid=1\r\n

cameraname=backdoor\r\n

eventId=2\r\n

eventpath=root.Event.E2\r\n

eventname=alarm\r\n

prebufferstartframe=456213:00\r\n

```

prebufferstarttime=98992000:00\r\n
eventstartframe=456345:00\r\n
eventstarttime=98992501:00\r\n
eventstopframe=456287:34\r\n
eventstoptime=98992500:34\r\n
postbufferstopframe=456288:34\r\n
postbufferstoptime=98992501:34\r\n
priority=2\r\n
ongoing=no\r\n
secured=no\r\n
startquality=3\r\n
stopquality=4\r\n
\r\n

```

Example 2: Search and list recordings of event 6 from camera 1:

`http://myserver/axis-cgi/view/reclist.cgi?camera=1&event=6`

Example 3: List all available recordings:

`http://myserver/axis-cgi/view/reclist.cgi`

[TOC](#)

5.7.2 Play recordings

With this interface recorded images can be replayed from the server.

Method: GET

Syntax:

```

http://<servername>/axis-cgi/view/player.cgi[?<parameter>=<value>
[&<parameter>=<value> ...]]

```

parameters and values listed below.

To make the player more effective when playing back and forward, stepping etc, it uses *recording id* and *session id*. *Recording id* are received by [list recording](#).

Use the recording id in the first player-call and then the session id returned in every player-response.

Example:

AXIS Server ->	-< Client
List all recordings	
<code>http://myserver/axis-cgi/view/reclist.cgi</code>	Server response containing info about all recordings

Play recording 24	
http://myserver/axis-cgi/view/player.cgi?recordingid=24	
	Server response containing sessionid=3 and images.
Play backward	
http://myserver/axis-cgi/view/player.cgi?sessionid=3&direction=backward	
	Server response containing sessionid=3 and images.

TOC

5.7.2.1 Recorded MJPG video request

Parameter/Syntax	Values	Description
recordingid=<int>	> 0 ¹	id of this recording. Got from another cgi-request and used for faster database search
sessionid=<int>	0 if no session, else > 0 ¹	id of this session. Returned in previous server response. Used for faster database search
camera	1, ... ²	Selects a camera.
info=<int>	1	Returns a description of this CGI-request. No image is generated.
starttime=<int>:<int>	Seconds since 1970: 1/100 seconds ³	The start time of the recorded stream 983453412:10
stoptime=<int>:<int>	Seconds since 1970: 1/100 seconds ³	The stop time of the recorded stream 983453412:10
startframe=<int>	frame id ³	785421:57
stopframe=<int>	frame id ³	78542157
direction=<string>	forward, backward	Get the stream of the images with the specified time interval in right/opposite direction. If omitted = forward

step=<int>	1,2,3,4,5...	Ex. If step is equal to 2 then it shows the frames #1,3,5,7,9 etc in the stream.
------------	--------------	----------------------------------------------------------------------------------

¹ Request must contain one of these. See example below.

² Product dependent. Check product specification.

³ Use either start/stop time or start/stop frame but not both.

If time is specified the first found subsection that matches the time (within the specified recording, if such is specified) will be played. Since there can be several subsections that matches the same time (due to changes in server time) there is no way of knowing which one was requested, hence the first match is always played.

Frame ID is used to directly identify a unique frame of video from a specific camera. This is independent of different recordings and changes in server time. The scale of the Frame ID is linear to real time.

Example 1: Play recording 5

```
http://myserver/axis-cgi/view/player.cgi?recordingid=5
```

Example 2: Play backward - (sessionid=3 was returned in previous [server response](#))

```
http://myserver/axis-cgi/view/player.cgi?sessionid=3&startframe=453412480
&stopframe=45370023&direction=backward
```

Example 3: Fast forward - (sessionid=3 was returned in previous [server response](#))

```
http://myserver/axis-cgi/view/player.cgi?sessionid=3&startframe=453412480
&stopframe=45370023&step=4
```

Example 4: Play time interval - (The first found matching interval for the camera is played)

```
http://myserver/axis-cgi/view/player.cgi?camera=4&starttime=982813227:00
&stoptime=982813720:00
```

TOC

5.7.2.2 Recorded JPEG image request

Parameter/Syntax	Values	Description
recordingid=<int>	> 0 ¹	id of this recording. Got by list recordings cgi-request and used for faster database search
sessionid=<int>	0 if no session, else > 0 ¹	id of this session. Returned in previous server response. Used for faster database search
camera	1, 2, 3, 4	Selects a camera.
info=<int>	1	Returns a description of this CGI-request. No image is generated.

time=<int>:<int>	Seconds since 1970: 1/100 seconds ²	The time of the recorded image: 983453412:10
frame=<int>	frame id ²	78542157
direction=<string>	forward, backward	Get the next/before image from the one specified here by the time
step=<int>	0, 1	If step=0, get the frame with given timestamp (snapshot) If step=1, get the frame next to given timestamp (step forward/backward)

¹ Request must contain one of these.

² Use either time or frame but not both.

Example 1: Get snapshot of specific camera and time

```
http://myserver/axis-cgi/view/player.cgi?sessionid=3&frame=98345380
&direction=forward&step=0
```

Example 2: Get next picture

```
http://myserver/axis-cgi/view/player.cgi?sessionid=3&frame=98345380
&direction=forward&step=1
```

TOC

5.7.3 Recorded MJPG video response

When a recorded MJPG video is requested, the server returns a continuously flow of jpeg files. The content type is "multipart/x-mixed-replace" and each image is preceded of data about the image and ends with a boundary string <boundary>.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<image>
```

where the proposed <boundary> is

myboundary

and the returned <image> field is

```
x-axis-dvr-session-id: <sessionid>\r\n
```

```

x-axis-dvr-frame-time: <time>\r\n
x-axis-dvr-frame-id: <frameid>\r\n
x-axis-dvr-frame-state: <state>\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
<image>

```

Meta tag	Description
x-axis-dvr-session-id: <id>	ID of the session
x-axis-dvr-frame-time: <sec>	Timestamp in string format epoch:hh
x-axis-dvr-frame-id: <id>	Frame ID.
x-axis-dvr-frame-state: <state>	Active alarm inputs when this picture was taken

Example:

```

HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
x-axis-dvr-session-id: 132
x-axis-dvr-frame-time: 1003445674:23
x-axis-dvr-frame-id: 563234:23
x-axis-dvr-frame-state: 3
Content-Type: image/jpeg\r\n
Content-Length: 15656\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
x-axis-dvr-session-id: 132
x-axis-dvr-frame-time: 1003445674:54
x-axis-dvr-frame-id: 563234:54
x-axis-dvr-frame-state: 3

```

```
Content-Type: image/jpeg\r\n
Content-Length: 14978\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
x-axis-dvr-session-id: 132
x-axis-dvr-frame-time: 1003445675:21
x-axis-dvr-frame-id: 563235:21
x-axis-dvr-frame-state: 3
Content-Type: image/jpeg\r\n
Content-Length: 15136\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
.
.
```

TOC

5.7.4 Preserve and release recordings

Note: This request requires Administrator or Operator access.

This request changes the status on a recording from released to preserved or vice versa.

A preserved recording is not effected of quality reduction but keeps the quality it had when preserved until it is released.

Syntax:

```
http://<servername>/axis-
cgi/operator/recadmin.cgi?camera=<camera id>&[preserve|release]=<recid>
```

Example 1: Preserve recording 4234 from camera 2:

```
http://myserver/axis-cgi/operator/recadmin.cgi?camera=2&preserve=4234
```

Example 2: Release recording 243592 from camera 1:

```
http://myserver/axis-cgi/operator/recadmin.cgi?camera=1&release=243592
```

TOC

5.7.5 Controlling recordings

This interface is used to start and stop recordings from a camera.

Method: GET**Syntax:**

```
http://<servername>/axis-cgi/view/record.cgi[?<parameter>=<value>
[&<parameter>=<value> ...]]
```

With the following parameters and values:

<parameter>=<value>	Values	Description
do=<string> ¹	start, stop	Start or stop recording.
camera=<int> ¹	1, ... ²	Selects a camera.

¹ Request must contain these. See example below.

² Product dependent. Check product specification.

Example 1: Start recording from camera 1

```
http://myserver/axis-cgi/view/record.cgi?do=start&camera=1
```

Example 2: Stop recording from camera 2

```
http://myserver/axis-cgi/view/record.cgi?do=stop&camera=2
```

[TOC](#)

5.8 I/O

The requests specified in the I/O section is supported by those camera products that have Input/Output connectors.

5.8.1 I/O control

5.8.1.1 Input

Input

Method: GET**Syntax:**

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

with the following parameters and values

<parameter>=<value>	Values	Description
check=<int>[,<int>, ...]	<id1>[,<id2>, ...] ¹	Returns the status of one or more inputs numbered <i>id1</i> , <i>id2</i> ,

See [5.2.1](#) for more information about server responses.

monitor=<int>[,<int>, ...] ²	< <i>id1</i> >[,< <i>id2</i> >, ...] ¹	Returns a multipart stream of "check" inputs (see return description below).
-----------------------------------------	---------------------------------------------------	------------------------------------------------------------------------------

¹ Number of inputs may differ between different camera and video servers. See product specification.

² Support for this parameter is product/release dependent.

Return: "monitor", i.e., multipart "check" parameter

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace; boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<monitor data>
```

where the proposed boundary <boundary> is

ioboundary

and the <monitor data> part is

```
Content-Type: text/plain\r\n
\r\n
<check data>
--<boundary>\r\n
```

and <check data> is

I0<n>:<action char>\r\n

and <n> is the I/O port number and <action char> is the action character described in the table above.

Note: The output can contain extra blank lines, i.e., extra \r\n within the sections.

Example: Monitor data on input ports 1, 2, 3, and 4

<http://myserver/axis-cgi/io/input.cgi?monitor=1,2,3,4>

Example: Monitor data on input port 1

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace; boundary=ioboundary\r\n
\r\n
\r\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
I00:/\r\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
I00:H\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
I00:\r\n
\r\n
\r\n
--ioboundary\r\n
```

```
Content-Type: text/plain\r\n
\r\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
\r\n
.
.
```

[TOC](#)**5.8.1.2 Output****Output****Method:** POST**Syntax:**

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

with the following parameters and values

<parameter>=<value>	Values	Description
check=<int>[,<int>, ...]	<id1>[,<id2>, ...] ¹	Returns the status of one or more outputs numbered <i>id1</i> , <i>id2</i> , See 5.2.1 for more information about server responses.
monitor=<int>[,<int>, ...] ²	<id1>[,<id2>, ...] ¹	Returns a multipart stream of "check" outputs (see return description below).
action=<string>	[<id> ¹]:<a>[<wait> <a> ...]	Sets the output relay <id> on or off and waits <wait> milliseconds. <id> = Output number. If omitted, output 1 is selected. <a> = Action character: / or \/ / = on, \ = off. <wait> = Delay in milliseconds.

¹ Number of outputs may differ between different camera and video servers. See product specification.² Support for this parameter is product/release dependent.**Example:** Set output 1 on

```
http://myserver/axis-cgi/io/output.cgi?action=1:/
```

Example: Set two 300 ms pulses with 500 ms delay between the pulses on output 1

```
http://myserver/axis-cgi/io/output.cgi?action=1:/300\500/300\
```

Example: Wait 1 second before setting output 1 on

```
http://myserver/axis-cgi/io/output.cgi?action=1:1000/
```

[TOC](#)

5.8.2 Virtual I/O control

5.8.2.1 Input

Input

Method: GET

Syntax:

```
http://<servername>/axis-cgi/io/virtualinput.cgi?<parameter>=<value>
```

with the following parameters and values

<parameter>=<value>	Values	Description
action=<string>	[<id>]:<a>	<p>Sets the virtual input <id> on or off.</p> <p><id> = Input number. If omitted, input 1 is selected.</p> <p><a> = Action character: / or \</p> <p>/ = on, \ = off.</p>

Example: Set virtual input 1 on

```
http://myserver/axis-cgi/io/virtualinput.cgi?action=1:/
```

[TOC](#)

5.9 Serial Port

The requests specified in the Serial Port section is supported by those camera products that have implemented a Generic driver.

5.9.1 Serial port control

Control serial port

Method: GET/POST

Syntax:

```
http://<servername>/axis-cgi/com/serial.cgi?<parameter>=<value>
[&<parameter>=<value>... ]
```

with the following parameter and values

<parameter>=<value>	Values	Description
port=<int>	1, ... ¹	The COM port is selected with this parameter.

write=<string> dataout ² =<string>	<bytestring>	<bytestring>: hex coded bytes with values of {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, a, b, c, d, e, f} Writes the specified data string to the selected serial port. Max string length: 128 bytes ¹ .
writestring=<string>	<url encoded string>	Writes the url encoded string to the selected serial port. Max string length: 128 bytes ¹ .
read=<int>	1, ...	Reads <i>n</i> bytes from the selected serial port. The returned data will be hexadecimal coded and placed between #s (e.g. #3A#)
wait=<int>	1 - 9	Specified in seconds. Used together with the "read" parameter. A read is terminated when the specified number of bytes is read or when the wait period has ended.
timeout=<int>	1 - 9000	Specified in milliseconds. Used together with the "read" parameter. A read is terminated when the specified number of bytes is read or the timeout has expired.

¹ Product dependent. Check product specification.

² Obsolete.

TOC

5.9.2 Open serial port

This CGI makes it possible to open the serial port using the HTTP protocol. Authentication is handled by the Web server.

- After an initial connect command from the client, the connection is kept alive until the client closes it.
- Several clients may be connected to the same serial port concurrently.
- After the connection has been setup, data sent from the client to the Web server is forwarded to the serial port, and incoming serial data is returned to all currently connected clients.

Syntax:

```
http://<servername>/axis-cgi/com/serial.cgi?<parameter>=<value>
[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
port=<int>	1, ... ¹	Select COM port.
camera=<int> unit=<int>	1, ... ¹	Selects the source camera or external unit. If omitted, and "port=" command is also omitted, the default camera/unit is used to determine the serial port to use.

connect=<string>	yes	Makes the server keep the connection open, and start acting as a link between the client and the serial port.
------------------	-----	---------------------------------------------------------------------------------------------------------------

¹ Product dependent. Check product specification.

[TOC](#)

5.10 PPP

The requests specified in the PPP section is supported by those camera products that have support for the PPP protocol.

5.10.1 Close PPP connection

Note: This request requires administrator access (administrator authorization).

Accessing this URL will force the Axis device to immediately close and disconnect any current PPP connection. When closing a PPP connection, all connections will be closed (dial-in, dial-out, callback).

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/closeppp.cgi[?<wait>]
```

with the following parameter and value

<parameter>	Value	Description
<wait>	<int>	Number of seconds from request is received until the PPP connection is closed. <i>Default:</i> 0

Return: close PPP connection

```
HTTP/1.0 204 No Content\r\n
```

[TOC](#)

5.10.2 Reset PPP connection maxtimer

Note: This request requires administrator access (administrator authorization).

Accessing this URL will reset the timer counting down the maximum time a device is allowed to be connected over a PPP connection.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/admin/resetppptimer.cgi
```

Return: reset PPP connection maxtimer

```
HTTP/1.0 204 No Content\r\n
```

[TOC](#)

5.11 Event

The requests specified in the Event section is supported by those camera products that have support for event handling.

5.11.1 Event administration

Note: This request requires Administrator or Operator access.

Used to administer the list of event entries.

Method: GET

Syntax:

```
http://<servername>/axis-cgi/operator/ev_admin.cgi?add
http://<servername>/axis-cgi/operator/ev_admin.cgi?delete=<entry>
```

5.11.1.1 Actions

These are the actions used to administer the events. Each event is given an entry identifier (*E0*, *E1* *E2*, etc) which later is used to delete the entry. The identifier of a deleted entry may be reused by a new entry.

Action/Syntax	Argument	Description
add		Add a new empty entry to list of events.
delete=<entry>	E0, E1, E2, ...	Delete an entry from list of events.

[TOC](#)

5.11.1.2 Server responses

The actions produce one of the following server responses:

Return: A successful *delete*:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

Return: A successful *add*:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
```

```
<entry> OK\r\n
```

Return: A failed request. Input values are probably incorrect:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Request failed: <error message>\r\n
```

Example: Adding new event entry

```
http://myserver/axis-cgi/operator/ev_admin.cgi?add
```

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
E7 OK\r\n
```

Example: Deleting event entry

```
http://myserver/axis-cgi/operator/ev_admin.cgi?delete=E7
```

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

[TOC](#)

5.12 Firewall

The requests specified in the Firewall section is supported by those camera products that have built-in firewall functionality.

5.12.1 Firewall administration

Note: This request requires Administrator access.

Used to administer the built-in firewall which restricts network access to the server. When enabled, only the hosts added through this interface will be able to access the server.

For each entry added you have to specify which services are to be accessible. This is done by parameters where you can specify a specific service or a range of port numbers for custom services.

Method: GET

Syntax:

```

http://<servername>/axis-cgi/admin/fwall_admin.cgi?list

http://<servername>/axis-cgi/admin/fwall_admin.cgi?enable[&<option> ...]

http://<servername>/axis-cgi/admin/fwall_admin.cgi?disable[&<option> ...]

http://<servername>/axis-cgi/admin/fwall_admin.cgi?add[&<parameter>
[=<value>] ...][&<option> ...]

http://<servername>/axis-cgi/admin/fwall_admin.cgi?update=<entry>
[&<parameter>[=<value>] ...][&<option> ...]

http://<servername>/axis-cgi/admin/fwall_admin.cgi?delete=<entry>
[&<option> ...]

```

[TOC](#)**5.12.1.1 Actions**

These are the actions used to administer the firewall. Each added host is given an entry identifier (*F0*, *F1* *F2*, etc) which later is used to modify or delete the entry. The identifier of a deleted entry may be reused by a new entry.

Action/Syntax	Argument	Description
list		Produce a list of all settings and entries.
add		Add a new entry to list of accepted hosts.
delete=<entry>	F0, F1, F2, ...	Delete an entry from list of accepted hosts.
update=<entry>	F0, F1, F2, ...	Update an entry in list of accepted hosts.
enable		Enable the firewall. Blocks access from all except the added entries.
disable		Disables the firewall. Enables access from everyone.

[TOC](#)**5.12.1.2 Parameters**

Parameter/Syntax	Values	Description
host=<address> ¹	host name ² or numerical Internet address	Host to add or modify.
http		Opens ports for http access.
https		Opens ports for https access.

ftp		Opens ports for ftp access.
telnet		Opens ports for telnet access.
tcp=<int>-<int>	1024 - 65535	Opens specified TCP port range.
udp=<int>-<int>	1024 - 65535	Opens specified UDP port range.

¹ Required by action *add*.

²Address specified by host name requires that server has access to a Domain Name Server.

TOC

5.12.1.3 Options

Before modifying the settings of an enabled firewall the script will verify that the host that requested the action will still have access to the server once the action is performed. If the client uses a proxy then the proxy access will be verified instead.

If the verification fails, i.e. the host would have been blocked from the server, the action is not carried out and an error is returned. This prevents accidental lock out by the administrator, etc.

By setting the *force* option this verification is skipped and all actions are carried out regardless of their consequences.

Option/Syntax	Description
force	Do not verify host accessibility.

TOC

5.12.1.4 Server responses

The actions produce one of the following server responses:

Return: A successful *enable*, *disable*, *delete* or *update*:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

Return: A successful *add*:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<entry> OK\r\n
```

Return: A failed request. Input values are probably incorrect:

```
HTTP/1.0 200 OK\r\n
```

```
Content-Type: text/plain\r\n\r\nRequest failed: <error message>\r\n
```

Return: A failed verification by an *enable*, *add*, *delete* or *update* request. No settings have been changed. Use the *force* option to override the verification.

```
HTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\nVerification failed for IP <Numerical Internet address>\r\n
```

Return: A list request.

```
HTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\n[Firewall]\r\nenabled=<value>\r\n\r\n<entry data>\r\n<entry data>\r\n...
```

where each *<entry data>* is:

```
[<entry>]\r\nhost=<value>\r\nhttp=<value>\r\nhttps=<value>\r\nftp=<value>\r\ntelnet=<value>\r\ntcp=<value>\r\nudp=<value>\r\n\r\n
```

The parameter values has the following syntax:

Parameter	Values	Description
enabled	yes, no	States if the firewall is enabled or disabled.
host	host name ¹ or numeric Internet address	The host for this entry.
http	yes, no	States if host has HTTP access.
https	yes, no	States if host has HTTPS access.
ftp	yes, no	States if host has FTP access.
telnet	yes, no	States if host has TELNET access.
tcp	[1024-65535]-[1024-65535]	Specific TCP port range accessible by host.
udp	[1024-65535]-[1024-65535]	Specific UDP port range accessible by host.

¹Address specified by host name requires that server has access to a Domain Name Server.

Example: Adding new host with HTTP and FTP access, with verification

```
http://myserver/axis-cgi/admin/fwall_admin.cgi?add&host=10.13.18.20&http&ftp
```

Response:

```
HTTP/1.0 200 OK\r\n
```

```
Content-Type: text/plain\r\n
```

```
\r\n
```

```
F1 OK\r\n
```

Example: Listing entries

```
http://myserver/axis-cgi/admin/fwall_admin.cgi?list
```

Response:

```
HTTP/1.0 200 OK\r\n
```

```
Content-Type: text/plain\r\n
```

```
\r\n
```

```
[Firewall]\r\n
```

```
enabled="yes"\r\n
```

```
\r\n
```

```
[F0]\r\n
```

```
\r\n
```

```
host=10.13.18.1\r\n
```

```
http=yes\r\n
```

```
https=no\r\n
```

```
ftp=no\r\n
```

```
telnet=no\r\n
tcp=4000-4001\r\n
udp=\r\n
\r\n
[F1]\r\n
host=10.13.18.20\r\n
http=yes\r\n
https=no\r\n
ftp=yes\r\n
telnet=no\r\n
tcp=\r\n
udp=\r\n
\r\n
```

Example: Deleting entry, with failed verification

```
http://myserver/axis-cgi/admin/fwall_admin.cgi?delete=F1
```

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Verification failed for IP 10.13.18.20\r\n
```

Example: Deleting entry, overriding verification

```
http://myserver/axis-cgi/admin/fwall_admin.cgi?delete=F1&force
```

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

[TOC](#)