VAPIX® VERSION 3

Video Streaming API

AXIS COMMUNICATIONS

COPYRIGHT NOTICE

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

VAPIX® LICENSE AGREEMENT

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

1. GRANT OF LICENSE

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

2. COPYRIGHT

The INTERFACE and the INTERFACE DESCRIPTION are owned by Axis and are protected by copyright laws and international treaty provisions. Any use of the INTERFACE and/or 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.

Video Streaming API

Table of Contents

Table of Contents

Video Streaming API

1 Overview

1.1 Description

This document explains the basic arguments and parameters to send, receive and configure a video stream using HTTP or RTSP requests.

1.1.1 References

All VAPIX® references are available at:

http://www.axis.com/vapix

1.2 Unknown Arguments

If an unknown argument is requested, for example if an argument is misspelled it will be ignored by the built-in server in the Axis product. That means that no response feedback will be given.

2 Video Streaming Over HTTP

The HTTP-based video interface provides the functionality for requesting single and multipart images and for getting and setting internal parameter values. The image and CGI requests are handled by the built-in web server.

2.1 Prerequisites

2.1.1 Identification

Property: `Properties.API.HTTP.Version=3`
Firmware: 5.00 and later.

2.2 Common Examples

Example 1:
Check supported VAPIX® version.

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

Example 2:
Check supported resolutions.

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

Example 3:
Check supported image formats.

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

Example 4:
Check the default resolution of video source 1.

```
http://myserver/axis-cgi/imagesize.cgi?camera=1
```

Example 5:
Request a Motion JPEG video stream.

```
http://myserver/axis-cgi/mjpg/video.cgi
```

2.3 Image Resolution

By using the `imagesize.cgi` you can find out the real image resolutions depending on the video format (PAL/NTSC). You can also verify that desired resolution as well as resolution after rotation.

2.3.1 Image Resolution Request

The `imagesize.cgi` is used to retrieve the image resolution.

Access control: viewer
Method: GET/POST

Syntax:

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

With the following arguments and values:

Argument	Description
Any image argument.	See *2.7 Image Request Arguments* for image CGI arguments.

Example 6:
Check the default resolution of video source 1.

Request:

```
http://myserver/axis-cgi/imagesize.cgi?camera=1
```

Response:

```
image width = 720
image height = 576
```

Example 7:
Request a specific resolution with supplied parameters for video source 1.

Request:

```
http://myserver/axis-cgi/imagesize.cgi?resolution=QCIF&rotation=180
&squarepixel=1&camera=1
```

Response:

```
image width = 192
image height = 144
```

2.3.2 Image Resolution Response

When an image resolution is requested, the Axis product either returns a resolution value or an error.

2.3.2.1 Successful Request

The image height and width in pixels are returned after a successful request.

Response:

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

Body:

```
image width=<value>
image height=<value>
```

2.3.2.2 Failure

If an Axis product does not support the requested resolution a body text is returned with info. For example `<!-- Camera 1 not available. -->`.

Response:

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

Body (value for argument "camera" is specified):

```
<!-- [error message] -->
```

Body (value for argument "camera" is not specified):

```
[error message]
```

2.4 Video Status

This section only applies to video encoders. The `videostatus.cgi` is used to check the status of one or more video sources.

2.4.1 Video Status Request

Request the status information for the video sources. The number of video sources in an Axis product is defined by the parameter `ImageSource.NbrOfSources`.

Access control: viewer
Method: GET

Syntax:

```
http://<servername>/axis-cgi/videostatus.cgi?<argument>=<value>
```

With the following argument and values:

Argument	Valid values	Description
`status=<int>[[,<int>],...]`	1 ...[1]	Check status of the listed video sources.

1. Product-dependent.

2.4.2 Video Status Response

Get the status information for the video sources.

Response:

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

Body:

```
Video 1 = <information>
...
```

`<information>` could be either `video` or `no video`.

Example 8:
Request video status from video source 1, 2, 3 and 4.

```
http://myserver/axis-cgi/videostatus.cgi?status=1,2,3,4
```

Response:

Returned data after a successful request.

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

Body:

```
Video 1 = video
Video 2 = no video
Video 3 = no video
Video 4 = video
```

The response `no video` means that there is no analog video signal attached to the physical video input port.

2.5 Bitmap

Support for bitmap images is product-dependent. Use the following command to check supported image formats.

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

Response example:

```
properties.image.format=jpeg,mjpeg,h264,bitmap
```

The response shows that the Axis product in the example supports bitmap images.

2.5.1 Bitmap Image Request

The syntax `bitmap/image.bmp` is used to request a bitmap image.

Access control: viewer
Method: GET

Syntax:

```
http://<servername>/axis-cgi/bitmap/image.bmp
[?<argument>=<value><[&<argument>=<value>...]]
```

With the following arguments and values:

Argument	Description
Bitmap image arguments.	See *2.7 Image Request Arguments* for arguments.

Example 9:
Request a bitmap image from the default video source using default settings.

```
http://myserver/axis-cgi/bitmap/image.bmp
```

Example 10:
Request a bitmap image from video source 1 with resolution 320x240.

```
http://myserver/axis-cgi/bitmap/image.bmp?resolution=320x240&camera=1
```

2.5.2 Bitmap Image Response

When a bitmap image is requested, the Axis product either returns the specified bitmap image file or an error.

2.5.2.1 Successful Request

Successful response to a HTTP request.

Response:

HTTP Code: `200 OK`
Content-Type: `image/bitmap`
Content-Length: *<image size in bytes>*

Body:

```
<bitmap image data>
```

2.6 JPEG/Motion JPEG

The requests specified in this section refer to Axis products that are set to use JPEG and Motion JPEG encoding.

2.6.1 JPEG Image (Snapshot) CGI Request

The `jpg/image.cgi` is used to request a JPEG image (snapshot). A JPEG image (snapshot) should only be used when requiring less than 1 fps.

Access control: viewer
Method: GET

Syntax:

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

With the following arguments and values:

Argument	Description
JPEG image arguments.	See *2.7 Image Request Arguments* for arguments.

Example 11:
Request a JPEG image from video source 1 with resolution 320x240 and compression 25.

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

Example 12:
Request a JPEG image from video source 2 with the text My Camera displayed.

```
http://myserver/axis-cgi/jpg/image.cgi?
&text=1&textstring=My%20Camera&camera=2
```

2.6.2 JPEG Image Response

When a JPEG image is requested, the Axis product either returns the specified JPEG image or an error.

2.6.2.1 Successful Request

Successful response to a HTTP request.

Response:

HTTP Code: `200 OK`
Content-Type: `image/jpeg`
Content-Length: *<image size in bytes>*

Body:

```
<JPEG image data>
```

2.6.3 Motion JPEG Video CGI Request

The `mjpg/video.cgi` is used to request a Motion JPEG video stream with specified arguments. The arguments can be specified explicitly, or a predefined stream profile can be used. Image settings saved in a stream profile can be overridden by specifying new settings after the stream profile argument.

Access control: viewer
Method: GET

Syntax:

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

In addition to the arguments described in *2.7 Image Request Arguments, on page 13* `mjpg/video.cgi` accepts the follwing arguments.

Argument	Valid values	Description
`streamprofile=`*<string>*	*<stream profile name>*	Use a predefined stream profile. Supported stream profile names are stored in the `StreamProfile.S#.Name` parameters.
`duration=`*<int>*	An unsigned integer	Specifies for how many seconds the video will be generated and pushed to the client. `0`=unlimited.
`nbrofframes=`*<int>*	An unsigned integer	Specifies how many frames the Axis product will generate and push. `0`=unlimited.
`fps=`*<int>*	An unsigned integer	Using fps it is possible to specify the frame rate from the Axis product. `0`=unlimited.
General image arguments, see *2.7 Image Request Arguments*.		

Example 13:
Request a Motion JPEG video stream from video source 1 with resolution 320x240 and compression 25.

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

Example 14:
Request a Motion JPEG video stream from the default video source with frame rate 5.

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

Example 15:
Request a Motion JPEG video stream using the `myprofile` stream profile but with a lower resolution.

```
http://myserver/axis-cgi/mjpg/video.cgi?
streamprofile=myprofile&resolution=CIF
```

2.6.4 Motion JPEG Video Response

When a Motion JPEG video is requested, the Axis product either returns the specified Motion JPEG video or an error.

2.6.4.1 Successful Request

If the request was successful, the Axis product returns a continuous flow of JPEG images. The content type is `multipart/x-mixed-replace` and each image ends with a boundary string <*boundary*>.

Response:

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

Body:

```
--<boundary>
<image>

--<boundary>
<image>
```

Where the returned <*image*> field is:

```
Content-Type:  image/jpeg
Content-Length:  <image size in bytes>

<JPEG image data>
```

2.7 Image Request Arguments

The following arguments and values can be used in JPEG, Motion JPEG or bitmap CGI requests. Unless overridden by a argument it is the default values as configured via the GUI (or `param.cgi`) that decides the characteristics of the image or video.

Argument	Valid values	Description
`resolution=`<*string*>	A string[1]	Resolution of the returned image. For supported resolutions, check in parameter `Properties.Image.Resolution`.
`camera=`<*string*>	`1 ...` `quad`	Selects the video source. If omitted the default value `camera=1` is used. This argument is only valid for Axis products with more than one video source. That is cameras with multiple view areas and video encoders with multiple video channels.
`compression=`<*int*>	`0 ... 100` [1]	Adjusts the compression level of the image. Higher values correspond to higher compression, that is lower quality and smaller image size. Note: This value is internally mapped and is therefore product-dependent.

`rotation=<int>`	`0` `90`[1] `180`[1] `270`[1]	Rotate the image clockwise. The number of rotation alternatives in an Axis product is defined by the parameter `Properties.Image.Rotation`.
`squarepixel=<int>`	`0` `1`	Enable/disable square pixel (aspect ratio) correction. If the parameter is set to `1` the Axis product will adjusts the aspect ratio to make it appear as intended.

1. Product/release-dependent.

Note

For arguments concerning overlays refer to the Overlay document available at:

http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php

3 RTSP API

RTSP (Real Time Streaming Protocol) is a control protocol for media streams delivered by a media server. RTSP can be considered a "remote control" providing commands such as play and pause. In addition, RTSP API provides parameters controlling media stream properties such as resolution, compression, video bit rate and audio as well as parameters controlling the image settings.

Please refer to the release notes for the actual product for compliance information.

The RTSP server in the Axis products is based on RFC 2326 *Real Time Streaming Protocol (RTSP)*, RFC 4566 SDP: *Session Description Protocol* and RFC 3550 *RTP: A Transport Protocol for Real-Time Applications*.

When streaming both video and audio the audio and video can be synchronized by using RTP timestamps as described in RFC 3550.

3.1 Prerequisites

3.1.1 Identification

Property: `Properties.API.RTSP.Version=2.01` and later
Property: `Properties.API.RTSP.RTSPAuth=yes`

3.2 RTSP Commands

The RTSP API provides several commands for media stream control.

3.2.1 Request Syntax

Syntax:

```
COMMAND rtsp://<servername>/axis-media/media.amp
[?<parameter>=<value>[&<parameter>=<value>...]]  RTSP/1.0<CRLF>
Headerfield1:  val1<CRLF>
Headerfield2:  val2<CRLF>
...
<CRLF>

[Body]
```

`COMMAND` is any of `DESCRIBE`, `SETUP`, `OPTIONS`, `PLAY`, `PAUSE`, `TEARDOWN`, `SET_PARAMETER` or `GET_PARAMETER`. Lines are separated with Carriage Return and Line Feed (`CRLF`).

Supported RTSP URL parameters and their values are listed in section *3.4 Parameter Specification RTSP URL*.

Note

RTSP requests always contain the absolute URL.

The following header fields are accepted by all commands. Some commands accept or require additional header fields:

Header Field	Description
`Authorization`	Authorization information from the client.
`CSeq`	Request sequence number.
`Session`	Session identifier (returned by the Axis product in `SETUP` response).
`Content-Length`	Length of content.

Content-Type	The media type of the content.
User-Agent	Information about the client that initiates the request.
Require	Query whether an option is supported. Unsupported features are listed in the Unsupported header field. See example 2 in section *3.2.10 RTSP SET_PARAMETER*.

3.2.2 Response Syntax

Syntax:

```
RTSP/1.0 <Status Code> <Reason Phrase> <CRLF>
Headerfield1:  val3<CRLF>
Headerfield2:  val4<CRLF>
...

[Body]
```

The first response line contains a status code and a reason phrase indicating the success or failure of the request. The status codes are described in RFC 2326.

The following header fields can be included in all RTSP response messages:

Header Field	Description
CSeq	Response sequence number (matches the sequence number of the request).
Session	Session identifier.
WWW-Authenticate	Authentication from client requested.
Date	Date and time of the response.
Unsupported	Features not supported by the Axis product.

3.2.3 RTSP DESCRIBE

The DESCRIBE command is used to request an SDP description of the media stream(s). The Session Description Protocol (SDP) is described in RFC 2327.

The DESCRIBE request accepts the additional header field:

Header Field	Description
Accept	List of content types that client supports (application/sdp is the only supported type).

The response to the DESCRIBE command contains the additional header fields:

Header Field	Description
Content-Type	Type of content (application/sdp).
Content-Length	Length of SDP description.
Content-Base	If relative URLs are used in the SDP description, this is the base URL.

Example 1:

Request:

```
DESCRIBE rtsp://myserver/axis-media/media.amp
?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  0
User-Agent:  Axis AMC
Accept:  application/sdp
```

Response:

```
RTSP/1.0 200 OK
CSeq:  0
Content-Type:  application/sdp
Content-Base:  rtsp://myserver/axis-media/media.amp/
Date:  Wed, 16 Jul 2008 12:48:47 GMT
Content-Length:  847

v=0
o=- 1216212527554872 1216212527554872 IN IP4 myserver
s=Media Presentation
e=NONE
c=IN IP4 0.0.0.0
b=AS:50064
t=0 0
a=control:rtsp://myserver/axis-media/media.amp?videocodec=h264
&resolution=640x480
a=range:npt=0.000000-
m=video 0 RTP/AVP 96
b=AS:50000
a=framerate:30.0
a=transform:1,0,0;0,1,0;0,0,1
a=control:rtsp://myserver/axis-media/media.amp/trackID=1?videocodec=h264
&resolution=640x480
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; profile-level-id=420029;
sprop-parameter-sets=Z0IAKeKQFAe2AtwEBAaQeJEV,aM48gA==
m=audio 0 RTP/AVP 97
b=AS:64
a=control:rtsp://myserver/axis-media/media.amp/trackID=2
?videocodec=h264&resolution=640x480
a=rtpmap:97 mpeg4-generic/16000/1
a=fmtp:97 profile-level-id=15; mode=AAC-hbr;config=1408; SizeLength=13;
IndexLength=3;IndexDeltaLength=3; Profile=1; bitrate=64000;
```

3.2.4 SDP Media Attribute Transform

Depending on product model, the SDP file may contain a video media attribute `transform`. If the streamed video is rotated or mirrored from the image source, this video media attribute shows how the video stream is orientated in relation to the original image configuration. The orientation is described by a transformation matrix consisting of homogeneous coordinates for two-dimensional operations (a 3x3 matrix).

Syntax:

```
a=transform:<MATRIX>
```

The matrix is formatted using commas to separate columns and semicolons to separate rows.

Example 2:
A video stream which is rotated 90 degrees is described by:

```
a=transform:0,-1,0;1,0,0;0,0,1
```

3.2.5 RTSP OPTIONS

The `OPTIONS` request returns a list of supported RTSP commands. The command can be used to keep RTSP sessions alive by repeating the `OPTIONS` request at regular intervals. The session timeout time is specified by the timeout parameter returned from the `SETUP` command (see *3.2.6 RTSP SETUP*).

The response to the `OPTIONS` command contains the additional header field:

Header field	Description
`Public`	Specify the supported RTSP commands.

Example 3:
List supported commands. The asterisk (*) makes the request apply to the server and not to a particular URL.

Request:

```
OPTIONS * RTSP/1.0
CSeq:  1
User-Agent:  Axis AMC
Session:  12345678
```

Response:

```
RTSP/1.0 200 OK
CSeq:  1
Session:  12345678
Public:  DESCRIBE, GET_PARAMETER, PAUSE, PLAY, SETUP, SET_PARAMETER, TEARDOWN
Date:  Wed, 16 Jul 2008 12:48:48 GMT
```

Note

As indicated in the response, the `GET_PARAMETER` command is supported; there are however no parameters to retrieve.

3.2.6 RTSP SETUP

The `SETUP` command is used to configure the data delivery method.

The `SETUP` request requires an additional header field which is also included in the response:

Header field	Description

`Transport`	Specify how the data stream is transported. Supported variants are:
	`RTP/AVP;unicast;client_port=port1-port2`
	`RTP/AVP;multicast;client_port=port1-port2`
	`RTP/AVP/TCP;unicast`

If using unicast in combination with TCP, it is recommended to increase the size of the RTP packets to 64 000 bytes (from the standard 1500 bytes), provided that the client can accept larger packets. Also for unicast streaming over RTP/UDP it might be beneficial to increase the packet size if no packets are dropped. The packet size is changed using the following header field in the `SETUP` request:

Header field	Description
`Blocksize`	Request a specific media packet size. The packet size should be a positive decimal number measured in octets.

The response returns a session identifier that should be used together with the stream control commands (for example PLAY, PAUSE and TEARDOWN). If the session header includes the timeout parameter, the session will close after the timeout time unless explicitly kept alive. Session can be kept alive by sending RTSP requests to the Axis product containing the session identifier (for example OPTIONS, see *3.2.5 RTSP OPTIONS*) within the timeout time or by using RTCP messages. Reconfiguration of transport parameters is not supported.

Example 4:
The response to the first SETUP request returns the session identifier (Session) which is used in subsequent requests. The parameter trackID should be read from DESCRIBE and used in SETUP.

Request:

```
SETUP rtsp://myserver/axis-media/media.amp/
trackID=1?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  2
User-Agent:  Axis AMC
Transport:  RTP/AVP;unicast;client_port=20000-20001
```

Response:

```
RTSP/1.0 200 OK
CSeq:  2
Session:  12345678; timeout=60
Transport:  RTP/AVP;unicast;client_port=20000-20001;
server_port=50000-50001;ssrc=B0BA7855;mode="PLAY"
Date:  Wed, 16 Jul 2008 12:48:47 GMT
```

Example 5:

Request:

```
SETUP rtsp:///myserver//axis-media/media.amp/
trackID=2?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  3
User-Agent:  Axis AMC
Transport:  RTP/AVP;unicast;client_port=20002-20003
Session:  12345678
```

Response:

```
RTSP/1.0 200 OK
CSeq:  3
Session:  12345678; timeout=60
Transport:  RTP/AVP;unicast;client_port=20002-20003;
server_port=50002-50003;ssrc=D7EB59C0;mode="PLAY"
Date:  Wed, 16 Jul 2008 12:48:48 GMT
```

3.2.7 RTSP PLAY

The PLAY request starts (or restarts if paused) the data delivery to the client.

Note

When playing Motion JPEG via RTSP there is a resolution limit of 2040x2040 pixels.

The response to the PLAY command contains the additional header fields:

Header field	Description
Range	The play time period.
RTP-Info	Information about the RTP stream, including the sequence number of the first packet of the stream.

Example 6:

Request:

```
PLAY rtsp://myserver/axis-media/media.amp
?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  4
User-Agent:  Axis AMC
Session:  12345678
```

Response:

```
RTSP/1.0 200 OK
CSeq:  4
Session:  12345678
Range:  npt=0.645272-
RTP-Info:  url=rtsp://myserver/axis-media/media.amp/
trackID=1?videocodec=h264&resolution=640x480;seq=46932;
rtptime=1027887748, url=rtsp://myserver/axis-media/media.amp/
trackID=2?videocodec=h264&resolution=640x480;seq=3322;rtptime=611053482
Date:  Wed, 16 Jul 2008 12:48:48 GMT
```

Example 7:
Play the recording "myrecording".

Request:

```
PLAY rtsp://myserver/axis-media/media.amp?recordingid="myrecording" RTSP/1.0
CSeq:  4
User-Agent:  Axis AMC
Session:  12345678
```

3.2.8 RTSP PAUSE

The PAUSE request is used to temporarily stop data delivery from the Axis product. Use PLAY to restart data delivery.

Example 8:

Request:

```
PAUSE rtsp://myserver/axis-media/media.amp
?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  5
User-Agent:  Axis AMC
Session:  12345678
```

Response:

```
RTSP/1.0 200 OK
CSeq:  5
Session:  12345678
Date:  Wed, 16 Jul 2008 12:48:49 GMT
```

3.2.8.1 RTSP PAUSE On Live Stream

If `PAUSE` is requested during live streaming the data transmission will stop immediately. If `PLAY` later is requested the live steam starts on the latest sampled frame. That means that the client will lose the video during the time that the stream has been paused. The client is notified in the Range header which interval that will be streamed.

3.2.9 RTSP TEARDOWN

The `TEARDOWN` request is used to close the data delivery from the Axis product.

Example 9:

Request:

```
TEARDOWN rtsp://myserver/axis-media/media.amp
?videocodec=h264&resolution=640x480 RTSP/1.0
CSeq:  6
User-Agent:  Axis AMC
Session:  12345678
```

Response:

```
RTSP/1.0 200 OK
CSeq:  6
Session:  12345678
Date:  Wed, 16 Jul 2008 12:49:01 GMT
```

3.2.10 RTSP SET_PARAMETER

The `SET_PARAMETER` command is used to change session parameters, currently only I-frame request is supported. The command sets the `Renew-Stream` parameter to `yes`.

Note

`Renew-Stream` must be sent in the body. The corresponding `Renew-Stream` parameter in some firmware 4.xx products had to be sent in the header. See example 2 below.

Example 10:
Use of `SET_PARAMETER` in firmware 5.xx products. `Renew-Stream` is sent in the body.

Request:

```
SET_PARAMETER rtsp://myserver/axis-media/media.amp RTSP/1.0
CSeq:  7
Session:  12345678
Content-Type:  text/parameters
Content-Length:  19

Renew-Stream:  yes
```

Response:

```
RTSP/1.0 200 OK
CSeq:  7
Session:  12345678
Date:  Wed, 16 Jul 2008 13:01:25 GMT
```

Example 11:
In some older Axis products, I-frames were requested using `RenewStream:yes` in the *header*. To find out whether Renew-Stream should be sent in the header or the body, the following method is recommended.

Send a request with `Require` and `RenewStream:yes` in the header.

Request:

```
SET_PARAMETER rtsp://myserver/axis-media/media.amp RTSP/1.0
CSeq:  7
Session:  12345678
Require:  com.axis.parameters-in-header
RenewStream:  yes
```

If the request is successful (response `200 OK`), the stream is renewed. Else, the Axis product responds with `551 Option not supported` (below) and `RenewStream` should be sent in the body.

Response:

```
RTSP/1.0 551 Option not supported
CSeq:  7
Session:  12345678
Unsupported:  com.axis.parameters-in-header
Date:  Wed, 16 Jul 2008 13:01:24 GMT
```

Send a second request with `RenewStream:yes` in the body.

Request:

```
SET_PARAMETER rtsp://myserver/axis-media/media.amp RTSP/1.0
CSeq:  8
Session:  12345678
Content-Type:  text/parameters
Content-Length:  19

Renew-Stream:  yes
```

Successful response.

Response:

```
RTSP/1.0 200 OK
CSeq:  8
Session:  12345678
Date:  Wed, 16 Jul 2008 13:01:25 GMT
```

3.3 RTSP Over HTTP

RTSP can be tunnelled over HTTP. This might prove necessary in order to pass firewalls etc. To tunnel RTSP over HTTP, two sessions are set up; one GET (for command replies and stream data) and one POST (for commands). RTSP commands sent on the POST connection are base64 encoded, but the replies on the GET connection are in plain text. To bind the two sessions together the Axis product needs a unique ID (conveyed in the x-sessioncookie header). The GET and POST requests are accepted on both the HTTP port (default 80) and the RTSP server port (default 554).

Note

For further information information see http://developer.apple.com/quicktime/icefloe/dispatch028.html

Syntax:

```
http://<servername>/axis-media/media.amp
```

Supported methods are GET and POST.

Example 12:
GET request.

Request:

```
GET axis-media/media.amp?videocodec=h264&audio=0 HTTP/1.0
x-sessioncookie:  123456789
```

Response:

```
HTTP/1.0 200 OK
Content-Type:  application/x-rtsp-tunnelled
```

Example 13:
POST request. There is no response from the Axis product.

Request:

```
POST axis-media/media.amp?videocodec=h264&audio=0 HTTP/1.0
x-sessioncookie:  123456789
Content-Length:  32767
Content-Type:  application/x-rtsp-tunnelled
```

After this request has been sent it is possible to send RTSP requests like below.

```
DESCRIBE rtsp://myserver/axis-media/media.amp?videocodec=h264 RTSP/1.0
CSeq:  14
User-Agent:  Axis AMC
Accept:  application/sdp
```

3.3.1 Network Parameters

The following parameters in the `Network.RTSP` group control RTSP authentication.

[Network.RTSP]

Parameter	Default values	Valid values	Access control	Description
`AuthenticateOverHTTP`	`no`[1]	`yes` `no`	admin: read	Perform a RTSP authentication when tunneling RTSP over HTTP. `yes`=The RTSP server requests authentication. This is made regardless if the HTTP-connection is authenticated or not. `no`=The RTSP server will not request authentication. It is assumed that the HTTP-connection already is authenticated.

1. Even if the current default behavior is not to require RTSP authentication when tunnelling through HTTP, this will probably change in the future. It is therefore strongly recommended to implement RTSP Digest authentication for all clients that use RTSP over HTTP.

3.4 Parameter Specification RTSP URL

RTSP API provides parameters for requesting media streams with specific properties and for image settings. The parameters are entered in the RTSP URL.

Video Streaming API

Syntax:

```
rtsp://<servername>/axis-media/media.amp
[?<parameter>=<value>[&<parameter>=<value>...]]
```

The following parameters are supported for H.264, MPEG-4 Part 2 and Motion JPEG streams:

Parameter	Valid values	Description
videocodec	h264 mpeg4 jpeg[1]	The selected video codec. Default: Product dependent; in order of priority: h264 , mpeg4 , jpeg.
streamprofile	A string	Name of a saved stream profile[2].
recordingid	A string	Name of a saved recording.
resolution	Product dependent	Specify the resolution of the returned image.
audio	0 1	Specify whether audio shall be available in the stream (for compatibility with applications without audio control). 0 = No audio. 1 = Audio. Default: 1
camera	1 ... quad[1]	Select the video source or the quad stream.
compression	0 … 100[1]	Adjust the compression level of the image. Higher values correspond to higher compression, that is lower image quality and smaller image size. Note: This value is internally mapped and therefore product-dependent.
colorlevel[1]	0 … 100[1]	Set the level of color or grey scale. 0 = Grey scale. 100 = Full color. Note: This value is internally mapped and therefore product-dependent.
color	0 1	Enable/disable color. 0 = Black and white. 1 = Color.
clock	0 1	Show/hide the time stamp. 0 = Hide. 1 = Show.
date	0 1	Show/hide the date. 0 = Hide. 1 = Show.
text	0 1	Show/hide the text. 0 = Hide. 1 = Show.
textstring	An URL-encoded string	Set the text shown in the image.
textcolor	black white	Set the color of the text shown in the image.

textbackgroundcolor	black white transparent semitransparent	Set the color of the text background shown in the image.
rotation	0 90 180 270[1]	Rotate the image clockwise.
textpos	0 1	The position of the string shown in the image. 0 = Top. 1 = Bottom.
overlayimage	0 1	Enable/disable overlay image. 0 = Disable. 1 = Enable.
overlaypos=<*int*>,<*int*> overlaypos=<*int*>x<*int*>[3]	Two unsigned integers	Set the x and y coordinates defining the position of the overlay image.
duration	An unsigned integer	Set the number of seconds the video will generate and push. 0 = Unlimited.
nbrofframes	An unsigned integer	Set the number of frames the Axis product will generate and push. 0 = Unlimited.
fps	An unsigned integer	Set the frame rate from the Axis product. 0 = Unlimited.
pull=<*bool*>	0 1	Optional parameter when using PLAY. 1 = Stream as fast as possible. Because the receiving part determines the transfer rate, this is only useful when tunneling RTSP over HTTP. 0 = Disabled. Default.

1. Values are product dependent. Check the product specification.
2. A stream profile is a set of video stream parameters (including videocodec) and is defined in the HTTP API or the web GUI. Supported stream profile names are stored in the StreamProfile.S#.Name parameters. It is possible override parameter values saved in a stream profile by specifying new values after the stream profile. See section for details.
3. Obsolete.

H.264 and MPEG-4 Part 2 streams support the following additional parameters:

Parameter	Valid values	Description
videobitrate	An integer	The rate (in kbits/s) at which video is requested. 0 = Variable bit rate. >0 = Constant bit rate with the given target bit rate. Default: 0
videomaxbitrate	An integer	Maximum bit rate (in kbits/s) for bit rate control. Default: 0 0 = Unlimited.

videobitratepriority	none framerate	The priority when rate control is used.
videokeyframeinterval	An integer	Corresponds to the GOV length setting in the web GUI. Default: 32, but product and codec dependent.

Motion JPEG streams support the following additional parameter:

Parameter	Valid values	Description
squarepixel	0 1	Enable/disable square pixel (aspect ratio) correction. If the parameter is set to 1 the Axis product will adjusts the aspect ratio to make it appear as intended.

3.5 Error Messages RTSP

The error messages for RTSP are described in RFC 2326.

3.6 RTCP

RTP Control Protocol (RTCP) is implemented according to the standard in RFC 3550.

4 Always Multicast

Always multicast means starting a multicast stream and letting it run continuously. Enabling always multicast reduces the latency when connecting to an Axis product. The always multicast streams enabled on the Axis product are presented by a Session Description Protocol (SDP). Using this information the client can choose to connect to the service.

4.1 Prerequisites

4.1.1 Identification

Property: `Properties.API.HTTP.Version=3`
Firmware: 5.40 and later.

4.2 SDP

The client makes a request according to the example below. The `camera` parameter specifies the desired video source on the Axis product.

To make a SDP request it is required that `Network.RTP.R0.AlwaysMulticastVideo=yes`.

4.2.1 Request SDP URL

Request:

```
http://myserver/axis-cgi/alwaysmulti.sdp?camera=1
```

4.2.2 Response SDP URL

The Axis product responds the request with a SDP. The SDP is protected by the HTTP authentication of the Axis product and requires admin access control.

Response:

```
v=0
o=- 1284464363092904 1284464363092904 IN IP4 axis
s=Multicast presentation
e=NONE
t=0 0
a=range:npt=0.000000-
m=video 50000 RTP/AVP 96
c=IN IP4 239.225.149.138/0
b=AS:50000
a=framerate:25.0
a=transform:1,0,0;0,1,0;0,0,1
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1; profile-level-id=420029;
sprop-parameter-sets=Z0IAKeKQFgJNgScFAQXh4kRU,aM48gA==
```

5 Stream Profiles

A stream profile is a set of video stream parameters suitable for different applications, devices or situations. Stream profiles can be used when retrieving a video stream from Axis products using the HTTP API, the RTSP API or the web GUI. All parameters that can be set in a video stream request can also be saved in a stream profile.

A few stream profiles are included at startup. The included stream profiles are designed according to basic requirements and can be customized by users with appropriate access rights. Users can easily create new stream profiles when needed. User-created profiles may also be removed.

Stream profiles are also used to define special image stream settings, for example for instant replay and events recording.

The stream profile API is an extension to the video stream CGI's Stream profiles parameters are added, updated, listed and removed using the parameter management CGI (param.cgi).

5.1 Prerequisites

5.1.1 Identification

Property: `Properties.API.HTTP.Version=3`
Firmware: 5.00 and later.

5.2 Common Examples

Example 1:
Add a new stream profile. In this example the new profile is the 5th stream profile so it will be referred to as `StreamProfile.S4`.

Request:

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

Response:

```
S4 OK
```

Example 2:
Add and configure a stream profile in one request. Here the profile is named `myprofile2` and the `Parameters` string is `videocodec=h264&resolution=4CIF&text=1&textstring=4CIF%20profile`. See *2.7 Image Request Arguments, on page 13* what arguments that could be used in the `Parameters` string.

Note

Characters in the `Parameters` string must be URL-encoded, so `resolution=CIF&text=1&textstring=CIF%20profile` becomes `resolution%3dCIF%26text%3d1%26textstring%3dCIF%2520profile`
The blank space is encoded as `%20`, the equal sign (=) as `%3d`, the ampersand (`&`) as `%26` and the percent sign is encoded as `%25`.

Request:

```
http://myserver/axis-cgi/param.cgi?action=add
&template=streamprofile
&group=StreamProfile
&StreamProfile.S.Name=myprofile2
StreamProfile.S.Description=My%204CIF%20profile
&StreamProfile.S.Parameters=videocodec%3dh264
%26resolution%3d4CIF%26text%3d1%26textstring%3d4CIF%2520profile
```

Response:

```
S5 OK
```

Example 3:
Configure a stream profile. In this example the profile is named `myprofile` and the `Parameters` string contains the following arguments: `resolution=CIF`, `text=1` and `textstring=CIF` profile. See *2.7 Image Request Arguments, on page 13* what arguments that could be used in the `Parameters` string.

Note

Characters in the `Parameters` string must be URL-encoded, so `resolution=CIF&text=1&textstring=CIF%20profile` becomes `resolution%3dCIF%26text%3d1%26textstring%3dCIF%2520profile`.
The blank space is encoded as `%20`, the equal sign (=) as `%3d`, the ampersand (`&`) as `%26` and the percent sign is encoded as `%25`.

Request:

```
http://myserver/axis-cgi/param.cgi?action=update
&StreamProfile.S4.Name=myprofile
&StreamProfile.S4.Description=My%20CIF%20profile
&StreamProfile.S4.Parameters=resolution%3dCIF%26text%3d1%26textstring%3dCIF%2520
profile
```

Response:

```
OK
```

Example 4:
List the parameters of a stream profile.

Request:

```
http://myserver/axis-cgi/param.cgi?action=list&group=StreamProfile.S5
```

Response:

```
root.StreamProfile.S5.Name=myprofile2
root.StreamProfile.S5.Description=My%204CIF%20profile
root.StreamProfile.S5.Parameters=videocodec%3dh264
%26resolution%3d4CIF%26text%3d1
%26textstring%3d4CIF%2520profile
```

5.3 Stream Profile Parameters

The parameters in the `StreamProfile` group control stream profile settings.

Note

In order to create a new dynamic parameter admin or operator access control is needed.

[StreamProfile.S#]

Template: streamprofile

Parameter	Valid values	Access control	Description

`Name`	`A-Z`
	`a-z`
	`0-9`
	`-, _`
		admin: read, write
		operator: read, write
		viewer: read
			The name of the stream profile used in the requests.
			
			Note: Each profile must have a unique name.

`Description`	A string.	admin: read, write operator: read, write viewer: read	User-friendly description of the profile.
`Parameters`	*<argument1>=<value1>* *&<argument2>=<value2>* `...`	admin: read, write operator: read, write viewer: read	List of arguments. See *2.7 Image Request Arguments* for complete list. Note: The characters must be URL-encoded.

Note

The # is replaced by a group number, for example `StreamProfile.S5`. The first group numbers are reserved for stream profiles included in the product

5.4 Motion JPEG Video Request

Saved stream profiles are convenient when retrieving Motion JPEG video streams through `video.cgi`. The value of a parameter saved in a stream profile can be overridden by specifying a new parameter value after the `streamprofile` argument.

Method: GET

Syntax:

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

With the following arguments:

Argument	Valid values	Description
`streamprofile=`*<string>*	Name of stream profile	The name of the stream profile. Supported stream profile names are stored in the `StreamProfile.S#.Name` parameters.
Additional arguments	See *page 13* for a complete list.	

Example 5:
Request, over HTTP, a Motion JPEG video stream configured according to the stream profile `myprofile`.

Request:

```
http://myserver/axis-cgi/mjpg/video.cgi?&streamprofile=myprofile
```

5.5 RTSP Request

Saved stream profiles are also convenient when requesting video streams using RTSP. The value of a parameter saved in the stream profile can be overridden in the RTSP request by specifying a new value after the streamprofile argument.

Syntax:

```
COMMAND rtsp://<servername>/axis-media/media.amp
?<argument>=<value>[&<argument>=<value>...]  RTSP/1.0
Headerfield1:  val1<CRLF>
Headerfield2:  val1<CRLF>
...
<CRLF>

[Body]
```

With the following arguments:

Argument	Valid values	Description
`streamprofile=<string>`	Name of stream profile	The name of the stream profile. Supported stream profile names are stored in the `StreamProfile.S#.Name` parameters.
Additional arguments	See *page 23* for a complete list.	

Example 6:
Stream profiles in RTSP requests. The value of a parameter saved in the stream profile can be overridden by specifying a new parameter value after `streamprofile`. Here, `myprofile2` (defined above) is used but the resolution is changed to 640x480.

Request:

```
DESCRIBE rtsp://myserver/axis-media/media.amp?
streamprofile=myprofile2&resolution=640x480 RTSP/1.0
CSeq:  0
User-Agent:  Axis AMC
Accept:  application/sdp
```

VAPIX® version 3
Video Streaming API

Ver. M4.2
Date: July 2013
Part No. 52937