# VAPIX® version 3

## Audio API

**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.

**Terms of Use**

The use of the AXIS VAPIX application programming interface (hereinafter referred to as "the INTERFACE" as further specified below, is subject to the terms and conditions of the License Agreement below. By using the INTERFACE and the written specification of the INTERFACE (hereinafter referred to as "the INTERFACE DESCRIPTION"), whether in whole or in part, you agree to be bound by the terms of the License Agreement.

# VAPIX LICENSE AGREEMENT

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

1. GRANT OF LICENSE

   Axis hereby grants to you the right to use the INTERFACE and the INTERFACE DESCRIPTION for the sole and limited purpose of creating, manufacturing and developing a solution that integrates any unit or portion included in the product range of Axis network cameras, Axis video servers, Axis video encoders and Axis video decoders (as defined by Axis at its discretion) and to market, sell and distribute any such solution.

2. COPYRIGHT

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

3. NO REVERSE ENGINEERING

   You may not reverse engineer, decompile, or disassemble the INTERFACE except to the extent required to obtain interoperability with other independently created computer programs as permitted by mandatory law.

4. TERMINATION

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

5. GOVERNING LAW

   This agreement shall be deemed performed in and shall be construed by the laws of Sweden. All disputes in connection with this agreement shall be finally settled by arbitration in accordance with the Rules of the Arbitration Institute of the Stockholm Chamber of

Commerce. The place of arbitration shall be Malmö, Sweden. The language of the proceedings, documentation and the award shall be English.

6. DISCLAIMER

6.1. THE INTERFACE AND THE INTERFACE DESCRIPTION ARE DELIVERED FREE OF CHARGE AND "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK AS TO THE USE, RESULTS AND PERFORMANCE OF THE INTERFACE AND THE INTERFACE DESCRIPTION IS ASSUMED BY THE USER/YOU. AXIS DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT AND PRODUCT LIABILITY, OR ANY WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE WITH RESPECT TO THE INTERFACE AND THE INTERFACE DESCRIPTION.

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

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

6.4. AXIS SHALL NOT BE LIABLE FOR LOSS OF DATA, LOSS OF PRODUCTION, LOSS OF PROFIT, LOSS OF USE, LOSS OF CONTRACTS OR FOR ANY OTHER CONSEQUENTIAL, ECONOMIC OR INDIRECT LOSS WHATSOEVER IN RESPECT OF USE OR DISPOSITION OF THE INTERFACE AND THE INTERFACE DESCRIPTION.

6.5. AXIS TOTAL LIABILITY FOR ALL CLAIMS IN ACCORDANCE WITH THE USE OF THE INTERFACE AND THE INTERFACE DESCRIPTION SHALL NOT EXCEED THE PRICE PAID FOR THE INTERFACE AND THE INTERFACE DESCRIPTION.

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

# Table of Contents

# 1 Overview

## 1.1 Description

Audio is supported by several of Axis network camera and video encoder models. Integrating audio with network video reduces the need and costs for extra cabling and allows audio to be transmitted to and/or from any point on the network, including remote locations.

Audio can be streamed on its own over HTTP or integrated with MPEG-4 or H.264 video streamed over RTP/RTSP. Audio configurations are global; the only allowed per-stream setting is audio enabling/disabling.

In addition to remote monitoring and recording, audio can be used in event handling. Events can be configured to trigger on audio alarms, which are generated when the sound level rises above or falls below a predefined alarm level.

**Note:** The audio/param.cgi from VAPIX 1 is removed. Use param.cgi instead (see examples below and the HTTP API version 3 specification).

### 1.1.1 Audio modes

Axis network video products can support some or all of the following audio modes

- **Full-duplex mode**
  Simultaneous two-way audio. Multiple clients can receive audio, but only one client at a time can transmit audio.
- **Half-duplex mode**
  Two-way audio, but only in one direction at a time.
- **Simplex – Speaker only mode**
  One-way audio where audio is transmitted from the client to the camera/video encoder.
- **Simplex – Microphone only mode**
  One-way audio where audio is transmitted from the camera/video encoder to the client. Multiple clients can receive audio at the same time.

### 1.1.2 Audio compression formats

Axis network video products can support all or some of the following audio compression formats

| Compression | MIME type | Bit rate (kbit/s) | Sample rate (kHz) |
|---|---|---|---|
| G.711 µ-law | audio/basic | 64 | 8 |
| G.726 | audio/G726-32 | 32 | 8 |
| | audio/G726-24 | 24 | 8 |
| AAC | audio/mpeg4-generic | 8,12,16,24,32 | 8 |
| | audio/mpeg4-generic | 12,16,24,32,48,64 | 16 |

AAC is usually transferred through RTP/RTSP; transfer via HTTP is supported but not recommended. AAC is not supported for multipart audio.

## 1.2 History

| Version | Date | Comment |
|---|---|---|
| 1.00 | 2008-Dec-15 | Initial version |

# 2 Prerequisites

## 2.1 Identification

**Property:** Properties.API.Version=3
Properties.Audio=yes
**Firmware:** 5.00 and above
**Product category:** Products with audio support (see the product's datasheet)

# 3 Common Examples

**Example 1:** Limit the maximum number of clients that can receive audio at the same time.

```
http://myserver/axis-cgi/param.cgi?action=update&Audio.MaxListeners=5
```

**Example 2:** Configure the audio source parameters.

```
http://myserver/axis-cgi/param.cgi?action=update
&AudioSouce.A0.Name=Dynamic%20Microphone
&AudioSource.A0.AudioEncoding=g726
&AudioSource.A0.InputType=mic
&AudioSource.A0.MicrophonePower=no
```

**Example 3:** Play a media stream (including both video and audio) using RTSP

```
PLAY rtsp://myserver/axis-media/media.amp?videocodec=h264
&audio=1 RTSP/1.0
CSeq: 2
User-Agent: Axis AMC
Session: 12345678
Authorization: Basic cm9vdDpwYXNz
```

**Example 4:** Request a singlepart audio stream using HTTP

```
http://myserver/axis-cgi/audio/receive.cgi?httptype=singlepart
```

**Example 5:** Successful response to a HTTP request. Here, singlepart audio data with G.711 µ-law compression is returned.
HTTP Code: 200 OK
Content-Type: audio/basic
Body:

```
<Audio data>
<Audio data>
<Audio data>
...
```

**Example 6:** Successful response to a HTTP request. Here, multipart audio data with G.726 32 kbit/s compression is returned.
HTTP Code: 200 OK
Content-Type: multipart/x-mixed-replace; boundary=myboundary
Body:

```
--myboundary
Content-Type: audio/G726-32
Content-Length: 256

<Audio data>
--myboundary
Content-Type: audio/G726-32
Content-Length: 256

<Audio data>
--myboundary
Content-Type: audio/G726-32
Content-Length: 256

<Audio data>
--myboundary
Content-Type: audio/G726-32
Content-Length: 256

<Audio data>
--myboundary
...
```

**Example 7:** Audio transmission. Singlepart audio using G.711 µ-law (authorization omitted).

```
POST /axis-cgi/audio/transmit.cgi HTTP/1.0
Content-Type: audio/basic
Content-Length: 9999999
Connection: Keep-Alive
Cache-Control: no-cache

<Audio data>
<Audio data>
<Audio data>
...
```

# 4 Parameters

## 4.1 Audio parameters

Common audio parameters used for all audio configurations.

**[Audio]**

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| DuplexMode | Product-dependent | full[1], half[1], get, post[1] | admin:rw operator:rw viewer:r | The audio mode. full = Full-duplex. Simultaneous two-way audio. half = Half-duplex. Two-way audio, but only in one direction at a time. get = Simplex. Retrieve audio from the server. post = Simplex. Send audio to the server. |
| MaxListeners | 10 or 20[1] | 0 ... 10 or 0 ... 20[1] | admin:rw operator:rw viewer:r | Maximum number of simultaneous audio clients (does not affect multicast delivery). |
| ReceiverBuffer[1] | 120 | 0 ... 9999 | admin:rw operator:rw | The receiving audio buffer size in milliseconds. |
| ReceiverTimeout | 1000 | 0 ... 9999 | admin:rw operator:rw | The receiving audio timeout in milliseconds. When the Axis video product is receiving audio data from a client, the session is terminated if no data is received in this time span. |
| NbrOfConfigs | Product-dependent | An unsigned integer | admin:r operator:r | The number of audio configurations. |
| DSCP | 0 | 0 ... 63 | admin:rw | The Differentiated Services Codepoint for audio Quality of Service (QoS). |

[1] Product/release-dependent. Check the product's Release notes.

## 4.2 Audio configuration

Audio configuration settings. Each audio configuration has its own parameter group.

**[Audio.A#]***

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| Enabled | no | yes,<br>no | admin:rw<br>operator:rw<br>viewer:r | Enable/disable the audio configuration. |
| HTTPMessageType | singlepart | singlepart,<br>multipart | admin:rw<br>operator:rw<br>viewer:r | How audio should be streamed. Some proxies require multipart streaming. |
| Name | | A string | admin:rw<br>operator:rw | Name of the configuration |
| Source | 0 | An integer[1] | admin:rw<br>operator:rw | The AudioSource.A# this Audio.A# configuration is connected to. |
| AlarmLevel | 50 | 0 ... 100 | admin:rw<br>operator:rw | Alarm level in percent of the maximum amplitude of the audio samples.<br>The alarm level is used in event setup. Events can be configured to trigger when the sound level rises above or falls below the alarm level. |
| AlarmResolution | 50 | 0 ... 100 | admin:rw<br>operator:rw | The length of the audio sample used for the audio alarm calculation. The parameter is expressed as percent of a block of 1024 samples, e.g. 50% corresponds to 512 samples. The actual sample time is the number of samples divided by the sample rate, e.g. 512 samples at 8 kHz correspond to 64 ms.<br><br>An audio alarm is generated when the mean level for a sample exceeds the AlarmLevel. A shorter AlarmResolution makes the alarm calculation more sensitive. |

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| AlarmLowLimit | 50[1] | 0 ... 10000 | admin:rw<br>operator:rw | The lowest configurable alarm limit (AlarmLevel=0%) in basis points (1/10000) of the maximum amplitude value. |
| AlarmHighLimit | 6500[1] | 0 ... 10000 | admin:rw<br>operator:rw | The highest configurable alarm limit (AlarmLevel=100%) in basis points (1/10000) of the maximum amplitude value. |

[1] Product/release-dependent. Check the product's Release notes.

* The # is replaced by a group number starting from zero, e.g. Audio.A0.

## 4.3 Number of audio sources

The number of audio sources.

**[AudioSource]**

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| NbrOfSources | 1[1] | An unsigned integer | admin:r<br>operator:r<br>viewer:r | The number of audio sources. |

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

## 4.4 Audio source settings

Audio source settings. Each audio source has its own parameter group.

**[AudioSource.A#]***

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| Name | Audio | A string | admin:rw<br>operator:rw | Name of the audio source. |
| AudioEncoding | aac[2] | g711[1],<br>g726[1],<br>aac[1] | admin:rw<br>operator:rw<br>viewer:r | The audio codec. |
| InputType | Hardware-dependent | internal[2],<br>mic,<br>line[2] | admin:rw<br>operator:rw | Where the audio shall be captured from and on which level it is. |

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| MicrophonePower | yes | yes,<br>no | admin:rw<br>operator:rw | Enable/disable power on the audio input connector. |
| InputGain | 0 | mute,<br>a number[3] | admin:rw<br>operator:rw | Gain (in dB) for sound sent from the server. |
| OutputGain[2] | 0 | mute,<br>an integer[3] | admin:rw<br>operator:rw | Gain (in dB) for sound sent to the server. |
| SampleRate | Hardware-dependent | 8000[2],<br>16000[2] | admin:rw<br>operator:rw | Clock rate (in Hz) for the audio sampling. |
| BitRate | Encoder-dependent | Encoder-dependent[4] | admin:rw<br>operator:rw | The output bit rate (in bits per second). |
| AudioSupport | yes | yes,<br>no | admin:rw<br>operator:r<br>viewer:r | Enable/disable audio from this audio source.<br>If the audio source is turned off with this parameter, no audio will be transmitted even if Audio.A#.Enabled=yes |

[1] Product-dependent. Check the corresponding Properties parameter.
[2] Product/release-dependent. Check the product's Release notes.
[3] Product/release-dependent.
[4] See the table in section 1.1.2

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

## 4.5 Properties parameters

**[Properties.Audio]**

| Parameter | Default value | Valid values | Security level | Description |
|---|---|---|---|---|
| Audio | Product-dependent | yes,<br>no | admin:r<br>operator:r<br>viewer:r | The product has audio support. |
| Format | g711, g726, aac[1] | A string | admin:r<br>operator:r<br>viewer:r | Comma-separated list of supported audio formats. |

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

# 5 HTTP API

Supported MIME types for audio

| MIME | Description |
|---|---|
| audio/basic | G.711 µ-law 64 kbit/s |
| audio/G726-32 | G.726 32 kbit/s |
| audio/G726-24 | G.726 24 kbit/s |
| audio/mpeg4-generic | AAC |

AAC is usually transferred through RTP/RTSP; transfer via HTTP is supported but not recommended. AAC is not supported for multipart audio.

## 5.1 Audio data request

Request a singlepart or multipart audio stream.

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

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

with the following argument and values

| Argument | Valid values | Description |
|---|---|---|
| httptype=<*string*> | singlepart, multipart | Choose streaming method. Some proxies require multipart streaming.<br>**Default**: As defined by the parameter Audio.A#.HTTPMessageType |

## 5.2 Singlepart audio data response

**1. Successful request**
If the request was successful, the server returns a continuous flow of audio packets. The content type is only set at the beginning of the connection. When the connection is up and running audio packets will come one after another without any extra information between the packets.

**Return**
**HTTP Code**: 200 OK
**Content-Type**: <*audio MIME*>
**Body:**

```
<audio data>
```

**2. Failure – Bad request**
If the specified parameter value is invalid, the server returns 400 Bad Request.

**Return**
**HTTP Code**: 400 Bad Request
**Body:**

```
<body>
```

## 5.3 Multipart audio data response

**1. Successful request**
If the request was successful, the server returns a continuous flow of audio packets. The content type is "multipart/x-mixed-replace" and each audio packet ends with a boundary string. The message body contains a block of binary data. The content length provides the size of each block of coded audio which varies for different codecs: G.711 has 512 bytes block size, G.726 32 kbit/s has 256 bytes and G.726 24 kbits/s has 192 bytes. AAC is not supported.

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

```
--<boundary>
<audio>

where the returned <audio> field is

Content-Type: <audio MIME>
Content-Length: <audio packet size>

<Audio data>
--<boundary>
<audio>
```

**2. Failure – Bad request**
If the specified parameter value is invalid, the server returns 400 Bad Request.

**Return**
**HTTP Code**: 400 Bad Request
**Body:**

```
<body>
```

## 5.4 Transmit audio data

Transmit a singlepart audio data stream.

**Security level:** viewer
**Method**: POST
**Syntax:**

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

There are no arguments to this CGI.

# 6 Audio in the RTSP API

Media streams transmitted over RTSP contain audio by default. See the VAPIX® RTSP API documentation for more information.

# 7 References

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

VAPIX® HTTP API version 3

VAPIX® Parameter Specification

VAPIX® RTSP API