



TECHNICAL DOCUMENT

Axis Technology Platform Migration Guide

Migration from 4.xx to 5.xx

Created: 29 July 2008
Last updated: 23 January 2009
Rev: 1.04

TABLE OF CONTENTS

Introduction	4
Important Note	4
Document History	5
1 Support for VAPIX 1 removed	5
2 New Streaming Architecture	6
2.1 Multi-streaming	6
2.1 Streaming protocols	6
2.1.1 ISMA standard	7
2.1.2 Multicasting	7
2.2 RTSP Server	8
2.2.1 RTSP Authentication	8
2.2.2 The media.amp interface supports parameters	8
2.3 RTP Packetization and RTP tunneling	9
2.4 Audio streaming	10
2.5 H.264 licensing	11
2.6 Summary of media streaming actions	11
3 New Time Format	11
4 Events Handling	12
4.1 Video buffering	14
5 Security	14
5.1 Digest Authentication	15
6 Configurable I/O Ports	15
7 Error Handling – HTTP Status Codes	17
8 Removed and Obsolete APIs	17

9	Axis H.264 Migration Experiences	18
	Step 1 Set up a HTTP-GET tunnel	18
	Step 2 H.264-specific information in the SDP data	18
	Step 3 Support H.264 de-packetization	19
	Step 4 Identify H.264 frame types	19
	Step 5 Enable “keep-alive”	19
10	Axis Development Tools	20
11	References	20

Introduction

This document is intended as a guide for application developers and describes how to migrate to Axis platform 5.xx with emphasis on utilizing the new H.264 standard, the most efficient video compression technique available today. The reader is presumed to have prior knowledge of Axis platforms, Axis platform programming and the H.264 standard.

Read more about H.264 in the white paper "*H.264 video compression standard. New possibilities within video surveillance.*" published by Axis Communications.

Firmware 5.xx was developed to take full advantage of the functionality enabled by Axis novel chip technology, the ARTPEC-3 and ARTPEC-B chips. The technical development has been driven by requirements from Axis' partners and customers including requests for a more flexible streaming architecture, increased performance and improved security. The new and improved features required major changes in VAPIX.

The first part of this document discusses changes in the streaming architecture and in the HTTP and RTSP APIs. The second part is a case study showing how Axis implemented support for H.264 in AXIS Camera Station version 3.x.

The purpose of this guide is to enable application developers to quickly and smoothly integrate Axis new product generation with their own applications. For a quick release it suffices to implement changes marked required. Recommended and optional changes can wait to a later stage, but should eventually also be implemented since it is most likely that obsolete parameters and APIs will not be supported in the future.

More information about the APIs, parameter groups etc can be found in the VAPIX documentation.

Important Note

Axis Communications AB provides no guarantee that any of the information given in this document will work for any particular application or that the descriptions will be valid for future platforms, firmware or product versions.

Axis Communications AB can not and will not be held liable for any damage inflicted to any product as a result of the examples or instructions mentioned in this document.

Axis Communications AB reserves the right to make changes to this document and to platform and product specifications without prior notice.

Axis Communications AB provides no support for application development of any kind.

Please bear in mind that the flash chip has a maximum lifespan of about 100,000 writes. For this reason, writing temporary files to the flash memory should be avoided; use the RAM disk mounted on `/tmp` instead.

Document History

Version	Date	Comment
1.0	29-Jul-2008	Initial version
1.01	6-Nov-2008	Correction: Valid values for the Videoformat parameter Addition: Multipart streaming in audio/transmit.cgi not supported
1.02	16-Dec-2008	Correction: mpeg-4 to mpeg4 in obsolete paths to media.amp Addition: New section on Error handling (400 Bad Request)
1.03	18-Dec-2008	More removed and obsolete APIs. Parameters Input.NbrOfInputs and Output.NbrOfOutputs marked as obsolete.
1.04	23-Jan-2009	Addition: RTP tunneling over RTSP or HTTP

1 Support for VAPIX 1 removed

The HTTP API version 1 (VAPIX 1) is not supported in firmware 5.xx products. The 3 CGIs

```
getparam.cgi
setparam.cgi
audio/getparam.cgi
```

which were obsolete in version 2 are removed and should be replaced by `param.cgi`.

Examples

To request information about network parameters:

```
OLD
  http://<host>/axis-cgi/admin/getparam.cgi?Network
NEW
  http://<host>/axis-cgi/param.cgi?action=list&group=Network
```

To set the default image resolution to 320x240 pixels:

```
OLD
  http://<host>/axis-cgi/admin/setparam.cgi?Image.Resolution=320x240
NEW
  http://<host>/axis-cgi/param.cgi?action=update
    &Image.I0.Resolution=320x240
```

To request information about audio parameters:

```
OLD
  http://<host>/axis-cgi/audio/getparam.cgi
NEW
  http://<host>/axis-cgi/param.cgi?action=list&group=Audio
```

2 New Streaming Architecture

The ARTPEC-3/ARTPEC-B chip technology and a new streaming architecture have enabled new functionality in Axis network video products, in particular multi-streaming and megapixel resolution. In addition to several new features, Axis' partners and customers have requested increased flexibility in the architecture in order to smoothly integrate Axis' products in their own applications and adapt to different security surveillance and remote monitoring situations. Compliance with standard media formats and communications protocols, as well as performance and security improvements, are other factors driving Axis technology development.

Disclaimer

The new streaming architecture is based on new code and is NOT intended to be bug-compatible. API and behavior compatibility is maintained wherever possible.

2.1 Multi-streaming

Multi-streaming, simultaneous streaming of individually configured media streams, is the most important new functionality enabled by Axis new chip technology. The parameters of each media stream are configured individually, permitting different resolutions, bit rates and compression levels etc for each stream. A typical example is simultaneous streaming of

- One high quality, high bit rate, low latency H.264 stream for live video
- One medium quality, medium bit rate, high latency H.264 stream for recording
- One low bandwidth, high latency H.264 stream for mobile or WAN
- One MJPEG stream for legacy players

Multi-streaming with individually configured media streams gives full flexibility in designing a security surveillance or remote monitoring environment.

The maximum number of simultaneous media streams varies from 3 for the ARTPEC-B chip to 15-20 for ARTPEC-3. Streaming too many high-resolution, high bit-rate streams at the same time is not recommended since the total encoder performance limits image quality, frame rate etc. If the encoder becomes overloaded, the performance of all streams will be reduced.

Predicting the maximum number of simultaneous media streams with, for example, maintained resolution and unreduced frame rate is a non-trivial task. The performance is hardware-dependent and limited due to a range of co-working factors including

- Sensor interface speed (the bottleneck between the image sensor and the image processor)
- Sensor specification, for example light sensitivity
- Video scaler performance: the scaler is shared by MJPEG and H.264 and converts the sensor resolution to the requested resolution
- Codec performance: the hardware encoders are separate for MJPEG and H.264
- Main CPU performance
- External system memory: amount and bandwidth

For optimal performance Axis recommends to use the same codec, resolution, compression, overlays etc for each simultaneous stream. The estimated maximum number of simultaneous high-resolution streams for a specific product can be found in the product's datasheet and/or User's Manual.

2.1 Streaming protocols

Axis network video products support multiple combinations of streaming and transport protocols together with the available video and audio formats. The options are summarized in Figure 1, where yellow blocks indicate combinations that are new in

firmware 5.xx. Note that the available video and audio formats may vary between products.

Axis' products are designed with separate modules for data transfer and network connection. The separate modules permit easy switching between for example the TCP and UDP transport protocols.

The AAC format streamed via HTTP over TCP is non-standard.

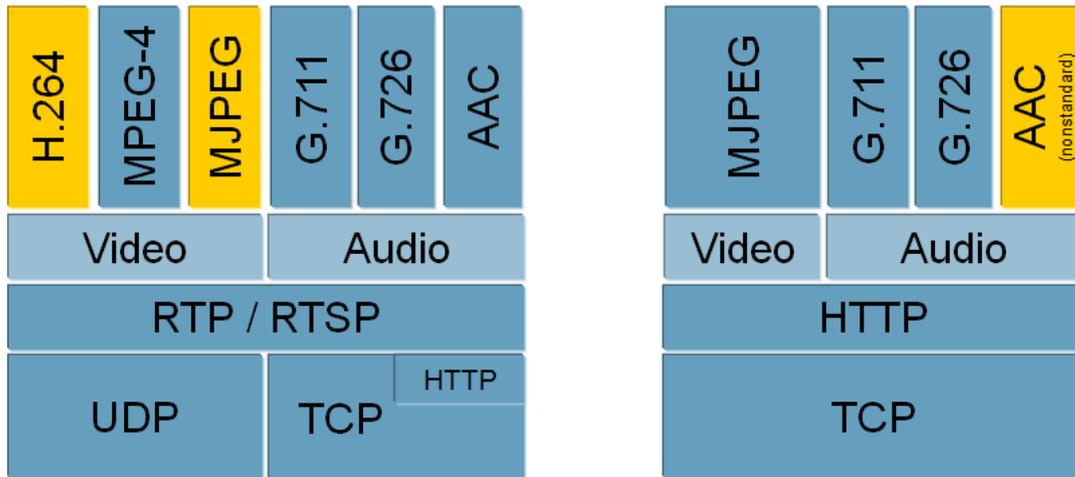


Figure 1: Video and audio formats, streaming and transport protocols supported by Axis firmware 5.xx. Yellow boxes indicate options new in 5.xx.

2.1.1 ISMA standard

ISMA, the Internet Streaming Media Alliance (<http://www.isma.tv>), publishes technical specifications to integrate multimedia standards and facilitate interoperability between different clients and servers when distributing media over a network.

Axis has chosen to follow the ISMA Implementation Specification v1.0 for MPEG-4 Part 2 and v2.0 for H.264. This means that RTP is used for media delivery, RTSP for media control and SDP for media presentation. For efficiency and simplicity, data from the codec is not formatted but delivered as an elementary stream (in raw format).

For video surveillance systems the H.264 Baseline Profile is the most suitable due to its low latency and bit-rate effectiveness combined with an acceptable complexity. The Level used together with the Baseline Profile in Axis' products is dependent on the parameter configuration of the media stream and is chosen automatically to give sufficient bit rate etc.

2.1.2 Multicasting

Though not recommended, in firmware generations preceding 5.xx multicasting could be started from the product setup with media data being transmitted whether or not there were viewing clients. It was also possible to fetch client-independent SDP data over HTTP. Firmware 5.xx removes these deficiencies and improves efficiency by requiring that all multicast sessions should be started properly. Data is only transmitted to clients that have explicitly requested access to the video stream. Multicasting sessions should be started using RTSP and client-dependent SDP data must be explicitly created. Clients set up a multicast route using IGMP.

2.2 RTSP Server

To accommodate multi-streaming and other new functionality, and for security improvements, firmware 5.xx introduces a number of changes in the RTSP server.

The API media.amp is accessed through the new URL

```
rtsp://<host>/axis-media/media.amp (NEW)
```

where <host> is the host name or IP address of the camera or video encoder. This path should be used for all codecs. The old URL

```
rtsp://<host>/mpeg4/<n>/media.amp (OLD)
```

where <n> is the video source number or quad stream (only for applicable products), is obsolete though still supported for backwards compatibility for MPEG-4 Part 2 streams in applicable products.

Axis RTSP implementation complies with the specifications described in RFC 2326. To follow the standard, the request line must always contain the complete (absolute) URL. This is necessary in order to support RTSP proxies. Relative paths are supported for backwards compatibility but generate a **syslog** warning.

Network connections are independent of RTSP sessions, such that, for example, multiple RTSP sessions over the same TCP connection are allowed. Remember, as of platform 4.4x, that an RTSP session must be explicitly opened and closed. To keep alive, sessions must be controlled by either an RTSP request (e.g. OPTIONS) or an RTCP message. To prevent the video from keeping transmitting, sessions must be closed properly using RTSP. The only exception is TCP transport where a live TCP connection is accepted as sign of an active channel. This is supported for backwards compatibility.

Required

Use the new URL `rtsp://<host>/axis-media/media.amp`

Recommendation

All non-standard behavior is considered obsolete functionality and will most likely be removed within a few years. To accelerate the migration to firmware 5.xx, start by implementing the required changes and make a release with those. As a second step, implement all clients to the full standard.

2.2.1 RTSP Authentication

With firmware 4.xx, authentication was only requested for the first RTSP packet of each TCP connection. To allow higher security, firmware 5.xx requires authentication of every RTSP packet regardless of the network transport protocol used.

The only exception is HTTP tunneling over TCP, where nothing more than HTTP authentication is required. This feature is supported for backwards compatibility and it is strongly recommended to implement full RTSP authentication. The backwards compatibility is controlled by the parameter `Network.RTSP.AuthenticateOverHTTP` with default value OFF. Using the default value, authentication is not required at all. Applications should be tested with both parameter values (on and off). To strengthen overall security, it is most likely that this feature will be removed or changed in the near future.

2.2.2 The media.amp interface supports parameters

To support multi-streaming and permit per-stream configuration, Axis has extended the media.amp interface to support parameters controlling resolution, compression, use of audio, codec etc. The new parameters are summarized in Table 1; complete parameter

specifications can be found in the VAPIX documentation. The interface should be accessed through a new URL

rtsp://<host>/axis-media/media.amp

where <host> is the host name or IP address of the camera or video encoder. The syntax for setting parameter values is similar to the one in the HTTP API, for example use

rtsp://<host>/axis-media/media.amp?videocodec=h264&audio=0

to select the H.264 codec and disable audio.

For MPEG-4 Part 2 and H.264 streams, Axis has as far as possible used parameters corresponding to the parameters already in use for MJPEG streams. Some parameters are specific to MPEG-4 Part 2/H.264, for example **videokeyframeinterval**. The **aspectratio** parameter, which is the ratio between a pixel's height and width, is only used by MJPEG. For the MPEG-4 Part 2 and H.264 formats, the aspect-ratio information is included in the VOL and VUI headers, respectively.

New parameters controlling MPEG-4 Part 2, H.264 and Audio			
Parameter	Default	Options	Description
videocodec	*	<i>h264</i> <i>mpeg4</i> <i>jpeg</i>	The selected video codec. *in order of priority, h264, mpeg4, jpeg depending on the codecs supported by the product
videobitrate	0	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
videomaxbitrate	0	integer	Maximum bit rate (in kbits/s) for bit rate control.
videobitratepriority		<i>none</i> <i>framerate</i> <i>quality</i>	The priority when rate control is used.
videokeyframeinterval	32*	integer	Corresponds to the GOV length setting in the web GUI. *Product and codec dependent.
audio	1	1,0	Specify whether audio shall be available in the stream (for compatibility with applications without audio control.

Table 1: New parameters in the media.amp interface. Complete parameter specifications can be found in the VAPIX documentation.

2.3 RTP Packetization and RTP tunneling

RTP packetization defines how the codec data is split up in RTP packets. Axis has implemented packetization for H.264 compliant with RFC 3984. Note that the packetizations for H.264 and MPEG-4 Part 2 are different.

Required

Applications have to implement compatible de-packetization. See RFC 3984.

When RTP is tunneled over RTSP or HTTP, the video streaming performance can be considerably improved if the Blocksize header in the RTSP SETUP is set to 64000 instead

of the default value 1400.¹ With this setting, the product will be able to handle more simultaneous video streams and, as a bonus, the network performance will improve. Note that the size of the tunneled RTP packets increases and that the client must be prepared to receive these larger packets. Axis has implemented this in AXIS Media Control and strongly recommends the same method for all clients that tunnel RTP over RTSP or HTTP.

2.4 Audio streaming

In contrast to video stream settings, the audio stream settings are global and do not permit per-stream configuration with the exception of enabling or disabling audio. There is no new audio functionality introduced in firmware 5.xx, but a few modifications in the handling of audio streaming, as summarized in Table 2.

The content-length parameter, which determines the size of the packets in the audio stream, used to have a fixed value but is now variable. This parameter is used in the HTTP multicast/singlepart audio header.

Audio forwarding is no longer supported and the corresponding parameters are obsolete. Transmitting multipart audio streams using `audio/transmit.cgi` is no longer supported. AAC Multipart is not supported.

Audio streaming changes	
New Parameter	
AudioSource.A#.AudioSupport	Controls the audio source. Requires administrator rights. If the audio source is turned off with this parameter, no audio will be transmitted even if Audio.A#.Enabled=yes .
Obsolete parameters	
Audio.NbrOfConfigs	Still exists but has no function.
Audio.ForwardOnPost	
Audio.ConnectedClients	
Audio.ConnectedPostClients	
Removed functionality	
Audio forwarding	
Multipart streaming with <code>audio/transmit.cgi</code>	
Obsolete functionality	
camera	Parameter in <code>audio/receive.cgi</code>

Table 2: Audio streaming changes.

¹ The default value is set to 1400 because some clients have hard-coded buffers sizes that cannot handle larger packets.

2.5 H.264 licensing

H.264 is a licensed technology. The patent license portfolio is administrated by the private organization MPEG-LA (<http://www.mpegla.com>).

Required

To deploy the H.264 technology in commercial products, contact MPEG-LA for more information and complete license terms. Up to 100,000 licenses¹ per year can be used for free, but an agreement with MPEG-LA must always be signed.

2.6 Summary of media streaming actions

Required

- Implement H.264 streaming and compatible de-packetization.
- Sign agreement with MPEG-LA.
- Use the new media.amp URL.
- If RTP/RTSP/HTTP/TCP is the only protocol used, there are no more required changes.
- If using RTP/RTSP/TCP or RTP/UDP, implement RTSP authentication.

Optional

- Support per-stream configuration of resolution, compression, etc.
- Support RTP/UDP.
 - o Implies extra RTSP authentication for added security and standards compliance.
 - o Implement correct session handling for RTP/UDP (keep-alive, closing sessions properly).

Recommended

- Start with RTP/RTSP/HTTP/TCP if you already support this.
- RTP/RTSP/TCP and RTP/UDP can be left as a future update in many cases.
- Leave per-stream configuration to future update (without it, it works as before).
 - o Multi-streaming and per-stream configuration are major features and should eventually be implemented.
- If RTP is tunneled over RTSP or HTTP, set the Blocksize header in the RTSP SETUP request to 64000.

3 New Time Format

In firmware generations preceding 5.xx, internal system time was set to the local time zone, so local time was used everywhere; in the web interface, in user data, in protocols etc. This caused unwanted gaps and overlays in video data when switching between daylight saving time and normal time.

In the 5.xx generation, system time is synchronized with UTC/GMT time and consequently independent of time zones and daylight saving time. In the protocols (RTCP, RTSP, HTTP etc) the UTC time appears in either POSIX or EPOCH² format.

Local time is still used in the web interface, in text overlays and for scheduled events (time-controlled events). Axis network video products can be queried for local time, time

¹ The number of free licenses could change. Please check with MPEG-LA.

² EPOCH is the number of seconds since January 1, 1970, not counting leap seconds.

zones and daylight saving time. A new feature in 5.xx is the possibility to include the local time zone and time offset from UTC in the text overlay.

Time parameter changes are summarized in Table 3.

Required

The new time format must be considered if time from the video source is used. Backwards incompatible.

Time parameter changes	
Removed parameters	
<code>root.Time.UTCOffset</code>	
<code>root.Time.DST.*</code>	All except <code>DST.Enabled</code> (DST = daylight saving time)
<code>root.Time.NTP.Update</code>	
Changed parameters	
<code>root.Time.TimeZone</code>	Now in POSIX format. Contains information about GMT offsets and daylight saving time. (Chapter 8.3, The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition.)
New functionality in text overlay options	
<code>%Z</code>	Time zone name (or abbreviation)
<code>%z</code>	Time offset from UTC

Table 3: Time parameter changes.

4 Events Handling

Most of Axis network cameras and video encoders can be configured to perform certain actions, such as uploading images or activate output ports, when certain types of events occur. Events can be activated by triggers such as video loss, motion detection, signals from input ports, tampering etc or scheduled to occur during pre-programmed time periods. Support for triggers and actions vary between camera and video encoder models.

Event configurations are stored as dynamic parameter groups (created at runtime) which can be set up and configured via HTTP or through the web interface. In firmware 5.xx, some event parameters have been modified, replaced or moved to support the H.264 video format and to provide a simpler, more transparent interface. The parameter changes are summarized in Table 4 and Table 5.

For improved performance and reduced latency, the event script functionality is removed and replaced with an application. All event-handling information is loaded at startup instead of parsing the script every time an event occurs.

Required

Embedded applications using the event script have to be rewritten. Backwards incompatible.

Changes in the Events.E# parameter group			
Parameter	Default	Options	Description
Active			Removed.
VideoFormat	jpeg	jpeg mpeg4 h264	New. Format used for upload/recording.
FrameRate	25	integer (fps fpm fph)	New. Frame rate per second/minute/hour. The same frame rate is used for the pre- and post-buffers. fps/fpm/fph = frames per second/minute/hour
PreTriggerSize			Replaced by PreTriggerDuration .
PreTriggerInterval			Replaced by PreTriggerDuration .
PreTriggerUnit			Removed. No longer needed.
PreTriggerIntervalUnit			Removed. No longer needed.
PreTriggerDuration	0	unsigned integer	New. The size of the buffer in milliseconds.
PreTriggerDurationUnit	s	s m h	New. Unit for PreTriggerDuration (in the web interface).
PostTriggerSize			Replaced by PostTriggerDuration .
PostTriggerInterval			Replaced by PostTriggerDuration .
PostTriggerUnit			Removed. No longer needed.
PostTriggerIntervalUnit			Removed. No longer needed.
PostTriggerDuration	0	unsigned integer	New. The size of the buffer in milliseconds.
PostTriggerDurationUnit	s	s m h	New. Unit for PostTriggerDuration (in the web interface).
BestEffortDuration	0	unsigned integer	Changed from seconds to milliseconds.
BestEffortInterval			Removed. No longer needed.
CreateFolder			Moved to Actions.
FileName			Moved to Actions.
Suffix			Moved to Actions.
MaxSequenceNumber			Moved to Actions.

Table 4: General event parameters.

Changes in the Event.E#.Actions.A# parameter group			
FTP & HTTP		SMTP	
CreateFolder	Moved from Event	FileName	Moved from Event
FileName	Moved from Event	Suffix	Moved from Event
Suffix	Moved from Event	MaxSequenceNumber	Moved from Event
MaxSequenceNumber	Moved from Event		

Table 5: Event Actions parameter changes.

4.1 Video buffering

When a triggered event is activated, Axis network cameras and video encoders can upload internally stored images from the time periods preceding and following the event using the pre- and post-trigger buffers. New in firmware 5.xx is that post-buffering continues if a new trigger is activated during post-buffering. For example, if configured to include 30 seconds post-trigger buffering and a new event occurs 20 seconds after the first one, post-buffering continues for an additional 30 seconds.

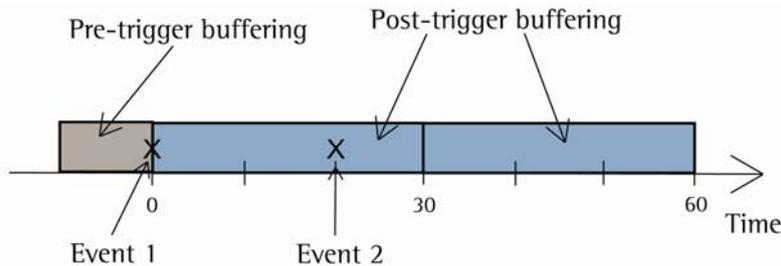


Figure 2: Video buffering. Post-buffering starts at Event 1 and is configured to continue for 30 seconds. When Event 2 occurs 20 seconds later, post-buffering continues and ends 60 seconds after Event 1.

Keep in mind that the image quality can be temporarily reduced when buffering is activate. Buffering has the same effect on the overall device performance as an additional video stream with the same configuration.

To improve performance and for easier setup, pre- and post-trigger buffers use the same frame rate. The `buffer/command.cgi` is removed. Parameter changes are described in Table 4 and Table 5.

5 Security

To strengthen overall security, provide better resistance against security exploits and prepare for future security updates, Axis has made major changes and improvements to the underlying security model. Most of these changes are not visible to the user and the required actions are limited to modifying URL paths.

The old security levels (1 for view, 4 for operator, 6 for administrator, etc.) are replaced by real access control lists. The actual access rights are determined from user identity and group membership instead of being controlled through access to a cgi. The default access rights for each user group are the same as in 4.xx.

In the VAPIX URLs, the **view/operator/admin** paths are removed. For example,
`http://<host>/axis-cgi/param.cgi` (NEW)
replaces

`http://<host>/axis-cgi/(view/operator/admin)/param.cgi` (OLD)

where **<host>** is the host name or IP address of the product.

For backwards compatibility, the old **view/operator/admin** paths and scripts still exist and map to the new paths and scripts.

5.1 Digest Authentication

HTTP Digest Authentication, as described in RFC 2617, is used for more secure authentication over both HTTP and RTSP. In Axis' products, Digest Authentication was introduced in firmware 4.xx and remains unchanged in 5.xx. Axis strongly recommends implementing digest authentication to increase overall security and for compatibility with systems and organizations not allowing basic authentication. It is most likely that digest authentication will be required by default in the near future.

6 Configurable I/O Ports

Most of Axis network cameras and video encoders have integrated digital input and output ports to connect external equipment such as detectors, lights, switches and alarm relays. The number of ports varies between product models.

Some products supporting the H.264 video format have configurable I/O ports to provide additional flexibility and adaption to demanding security surveillance and remote monitoring situations. In these products, the port direction can be set to input or output from VAPIX or the web interface.

To accommodate the configurable ports, a new API, **port.cgi**, replaces the old **input.cgi** and **output.cgi**. The new API is accessed through

`http://<host>/axis-cgi/io/port.cgi` (NEW)

where **<host>** is the host name or IP address of the camera or video encoder. The parameters in **port.cgi** are summarized in Table 6. The **input.cgi** and **output.cgi** are obsolete but supported for backwards compatibility.

The new **port.cgi** contains the same functionality as the obsolete **input.cgi** and **output.cgi**. To simulate the activation of an input port, use **virtualinput.cgi**.

The Input and Output parameter groups are obsolete and replaced by the new IOPort parameter group. See Table 7.

Required

The new API **port.cgi** and the new parameter group must be considered in applications using input and/or output ports.

Parameters in port.cgi		
Parameter	Options	Description
check	<i>id1</i> , [<i>id2</i> ,...]	Returns the status (1 or 0) of one or more ports numbered <i>id1</i> , <i>id2</i> , ...
checkactive	<i>id1</i> , [<i>id2</i> ,...]	Returns the status (active or inactive) of one or more ports numbered <i>id1</i> , <i>id2</i> , ...
checkdirection	<i>id1</i> , [<i>id2</i> ,...]	New functionality! Returns the direction (input or output) of one or more ports numbered <i>id1</i> , <i>id2</i> , ...
monitor	<i>id1</i> , [<i>id2</i> ,...]	Returns a multi-stream part of "check" ports. Inputs and outputs must be monitored separately.
action	[<i>id</i>]: < <i>a</i> >[, < <i>wait</i> > < <i>a</i> >...]	Valid only for output ports. < <i>id</i> > = Port number, output1 is default < <i>a</i> > = action character: / or \ /=active, \=inactive < <i>wait</i> > = delay in milliseconds

Table 6: Parameters in port.cgi. The checkdirection parameter is new; the other parameters are taken from the obsolete input.cgi and output.cgi.

New parameter group IOPort				
Parameter	Default value	Valid values	Description	
IOPort.I#.Configurable	yes	yes, no	New. Check if the port is configurable. Read-only and product dependent.	
IOPort.I#.Direction	input	input, output	New. The port direction. Read-only for non-configurable ports.	
IOPort.I#.Input.Name	Input 1	string	Replaces Input.I#.Name	
IOPort.I#.Input.Trig	closed	closed, open	Replaces Input.I#.Trig	
IOPort.I#.Output.Name	Output 1	string	Replaces Output.O#.Name	
IOPort.I#.Output.Active	closed	closed, open	Replaces Output.O#.Active	
IOPort.I#.Output.Button	none	none, pulse, actinact	Replaces Output.O#.Button	
IOPort.I#.Output.PulseTime	0	integer	Replaces Output.O#.PulseTime	
Obsolete parameters				
Input.NbrOfInputs				
Output.NbrOfOutputs				

Table 7: I/O parameter groups. # is replaced by a group number starting from 0.

¹ Product dependent.

7 Error Handling – HTTP Status Codes

For CGIs that return images or audio, the server will return 400 Bad Request if an invalid value for a known URL option is used. For example, in

```
http://myserver/axis-cgi/mjpg/video.cgi?resolution=abc
```

“abc” is not a valid value so 400 Bad Request is returned for firmware 5.xx products. For firmware 4.xx products, the response is a 200 OK with an error message in the HTTP body.

8 Removed and Obsolete APIs

Removed

```
/axis-cgi/buffer/command.cgi
```

MPEG-4 in HTTP API

```
/mpeg4[<camera>]/video.sdp
```

```
/mpeg4[<camera>]/media.sdp
```

```
/axis-cgi/mpeg4/restart_stream.cgi
```

```
/axis-cgi/mpeg4_statistics.cgi
```

Use `/axis-media/media.amp`

VAPIX1 no longer supported

```
/axis-cgi/admin/getparam.cgi
```

```
/axis-cgi/admin/setparam.cgi
```

```
/axis-cgi/audio/getparam.cgi
```

Replaced by `/axis-cgi/param.cgi`

Obsolete

IO API

```
/axis-cgi/io/input.cgi
```

```
/axis-cgi/io/output.cgi
```

Replaced by `/axis-cgi/io/ports.cgi`

Image and Video

```
/bitmap[<camera>]/image.bmp
```

```
/jpg[<camera>]/image.jpg
```

```
/mjpg[<camera>]/video.mjpg
```

MPEG-4 API

```
/mpeg4/<n>/media.amp
```

Replaced by `/axis-media/media.amp`

Obsolete paths

```
.../view/...
```

```
.../operator/...
```

```
.../admin/...
```

9 Axis H.264 Migration Experiences

This section describes how Axis implemented support for H.264 in AXIS Camera Station 3.x and points out the standard documents where communications protocols, video and audio formats are described.

To enable basic H.264 streaming 5 steps were taken:

1. Set up a HTTP-GET tunnel to establish a connection to the camera
2. H.264-specific information in the SDP data
3. Support H.264 de-packetization of packetized RTP packets
4. Identify H.264 frame types
5. Enable “keep alive” to keep RTSP sessions connected

Step 1 Set up a HTTP-GET tunnel

Axis chose to establish the connection to the camera using RTSP over HTTP (the Apple QuickTime streaming technology). Axis design with separate modules for data transfer (RTSP) and network connection enables a smooth switch to other protocols; only a single module must be changed to make the system communicate over UDP instead.

To be able to re-use the architecture already present in existing products, the URL for H.264 was constructed similar to the one used for MPEG-4 Part 2

```
http://<host>/axis-media/media.amp?videocodec=h264 (NEW)
```

```
http://<host>/mpeg4/media.amp (OLD)
```

The form of the new URL is advantageous as the codec part is separated from the path, making updates, extensions and other modifications straightforward.

As in the 4.xx generation, the HTTP connection can be authenticated using either Basic or Digest Authentication. For better security and compatibility with systems not allowing basic authentication, Axis strongly recommends using digest authentication.

A separate HTTP-POST tunnel was used for sending commands to the camera.

Reference document

Apple QuickTime RTSP/HTTP streaming

Step 2 H.264-specific information in the SDP data

The SDP data contains new H.264-specific information. Below is an extract from the video media level part of the SDP data; red text indicates updated data.

```
m=video 0 RTP/AVP 96
b=AS:8000
a=framerate:30.0
a=control:trackID=1
a=rtpmap:96 H246/90000
a=fmtp:96 packetization-mode=1; profile-level-id=420032;
sprop-parameter-sets=Z0IAMuKQFAey, aM48gA==
```

The encoding name in the `rtpmap` attribute is changed to H246. This information is needed in order to interpret the information provided in the `fmtp` attribute.

The `fmtp` attribute contains media format specific parameters. The `packetization-mode` parameter is used by the client to determine which type of de-packetization to use. The two comma separated parameters contained in the `sprop-parameter-sets` parameter are base-64 encoded. One of these parameters is the sequence parameter (see ISO/IEC

14496 part 10, section 7.3.1 and 7.4.1) which contains the picture size and other data required to set up the decoder environment.

Reference documents

RFC 3984 (RTP Payload Format for H.264 Video)
ISO/IEC 14496 part 10, sections 7.3.1, 7.3.2.1, 7.4.1 and 9.1

Step 3 Support H.264 de-packetization

RTP packets containing video and audio data are sent in packetized form from the camera or video encoder and must be de-packetized before arriving at the decoder. After de-packetization the original bit stream is recovered and can be processed and sent to the decoder. The SDP data from Step 2 specifies packetization mode 1 which should be used to de-packetize the RTP packets. Note that packetization differs between MPEG-4 Part 2 and H.264.



Figure 3: De-packetization of RTP packets.

Reference documents

RFC 3984 sections 5.6 (single NAL unit packet) and 5.8 (fragmented NAL unit packet)

Step 4 Identify H.264 frame types

The H.264 standard defines a new frame type, the IDR¹ frame. An IDR I-frame resets the decoder completely so that subsequent P-frames cannot reference frames preceding the IDR-frame. A H.264 I-frame on the other hand contains a complete image but does not reset the decoder. This means that P-frames following an (non-IDR) I-frame can reference frames preceding the I-frame.

The NAL² unit type, included in the NAL header, was used to recognize IDR and non-IDR frames. Note that since Axis uses the H.264 Baseline Profile, P-frames can never refer to frames preceding an I-frame.

Reference document

ISO/IEC 14496 part 10, section 7.4.1 (the NAL unit type)

Step 5 Enable “keep-alive”

The final step was to ensure that the RTSP session is kept open so that clients continue to receive media streams. With the TCP connection from Step 1, implementing keep-alive is not necessary as the established TCP connection already ensures open sessions. In order to be able to easily switch transport protocol, it still makes sense to implement keep-alive. As discussed above, the RTSP module is completely separated from the underlying communications module.

¹ IDR = Instantaneous Decoding Refresh

² NAL = Network Abstraction Layer

Implementing keep-alive was straightforward and did not take long. The response from the RTSP SETUP command contains a `timeout` parameter (time in seconds). For example:

```
RTSP/1.0 200 OK
Cseq: 2
Session: 2059134171;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=52-53;mode="PLAY"
```

To keep the session alive the RTSP command OPTIONS is sent every 30 seconds (`timeout/2`). The OPTIONS command does not change the state of the camera or video encoder. The response from the camera is sent interleaved with the media data in the HTTP-GET channel.

Reference document

RFC 2326 (Real Time Streaming Protocol)

10 Axis Development Tools

The following tools support the H.264 media format.

Software development kits (SDK)

- AXIS Media Control SDK v 5.40
- AXIS Media Parser SDK v 3.x
- AXIS Camera Station SDK

Applications

- AXIS Camera Station v 3.x
- AXIS Virtual Camera v 3.x

11 References

Apple QuickTime RTSP/HTTP streaming

RFC 3984 (RTP Payload Format for H.264 Video)

<http://tools.ietf.org/html/rfc3984>

RFC 2326 (Real Time Streaming Protocol)

<http://tools.ietf.org/html/rfc2326>

RFC 2617 (HTTP Digest Authentication)

<http://tools.ietf.org/html/rfc2617>

ISO/IEC 14496 part 10, sections 7.3.1, 7.3.2.1, 7.4.1 and 9.1

ISMA standard v1.0 (MPEG-4 Part 2) and v2.0 (H.264)

The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition

H.264 video compression standard. New possibilities within video surveillance. White paper from Axis Communications AB.

http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf

VAPIX documentation including Parameter Specifications, Axis Development Guidelines

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