Axis Development Guidelines
Contents

Axis Development Guidelines .......................................................................................................... 1
1. Important Note .......................................................................................................................... 4
2. Axis video product integration with software applications ...................................................... 4
3. AXIS Camera Application Platform ........................................................................................ 4
4. Join Axis ADP program ............................................................................................................ 5
5. VAPIX® .................................................................................................................................... 5
   5.1. Obtaining video ............................................................................................................... 5
6. Parameter configuration ............................................................................................................. 6
7. Dynamic parameters ................................................................................................................. 6
8. Video motion detection ............................................................................................................. 7
9. Notification of camera events ................................................................................................... 7
   9.1. TCP notifications ............................................................................................................ 7
   9.2. HTTP upload and notifications ....................................................................................... 8
   9.3. Trigger data in the video stream .................................................................................... 8
   9.4. Dynamic event notification ........................................................................................... 13
10. PTZ configuration and control ............................................................................................... 19
11. Privacy masking and overlay images .................................................................................... 19
12. Supported image resolutions ................................................................................................. 19
13. Properties parameters ............................................................................................................ 19
14. Developer information .......................................................................................................... 20
15. Axis Windows SDKs and development tools ....................................................................... 20
   15.1. AXIS Media Control SDK .......................................................................................... 20
   15.2. AXIS Media Parser SDK .......................................................................................... 21
   15.3. AXIS Camera Management SDK .............................................................................. 21
   15.4. AXIS Joystick SDK .................................................................................................... 21
   15.5. AXIS Virtual Camera .................................................................................................. 21
   15.6. AXIS Custom Firmware Tool ..................................................................................... 22
   15.7. Distribution ................................................................................................................... 22
16. Hardware and supported features .......................................................................................... 22
17. Embedded development .......................................................................................................... 22
   17.1. Considerations ............................................................................................................. 22
   17.2. Shell scripting and custom web interface ........................................................................ 23
   17.3. Application development using C ................................................................................ 23
18. Axis tools ............................................................................................................................... 23
1. **Important Note**

Axis Communications AB provides no guarantee that any of the examples shown in this document will work for any particular application.

Axis Communications AB cannot and will not be held liable for damage caused to any device as a result of the examples or instructions presented in this document.

Axis Communications AB reserves the right to make changes to this document without prior notice.

Please bear in mind that the flash chip manufacturer estimates the number of writes to the flash chips to about 100,000. So writing a lot of temporary files to the flash memory should be avoided. Use the ram disk mounted on /tmp instead.

2. **Axis video product integration with software applications**

This document describes guidelines from Axis Communications AB for setting up and integrating Axis video products with a software application.

There are different ways whereby an application can interface with Axis video products:

- Using **VAPIX®** directly. See section 5 below
- **AXIS Camera Application Platform.** See section 3 below
- Using Axis Windows development tools

Axis development tools, such as AXIS Media Control and AXIS Media Parser, can be seen as wrappers around **VAPIX®**, hiding most of the complexity involved in the integration and offering an easier interface for most applications.

3. **AXIS Camera Application Platform**

AXIS Camera Application Platform is an open application platform that makes it possible for members of the ADP program to develop and provide applications for the world’s leading and broadest portfolio of network video products.

When you develop an application and it is certified as being compatible with Axis products, it is listed on Axis’ web site where it can be downloaded and installed on Axis network cameras and video encoders. AXIS Camera Application Platform includes the following:

- An SDK for third party development of embedded applications that includes: a compiler system, APIs to access on-board device functionality, documentation, example code, and a debugger and profiler tool

  Note that the embedded development SDK requires a complete GNU/Linux operating system.

- Optional copy protection service to ensure that all applications in an installation are properly licensed

- A compatibility tool to ensure that the downloaded application does not impact the performance of Axis network cameras or video encoders
4. Join Axis ADP program

The Axis Application Development Partner (ADP) program offers several Windows SDKs – such as for recording and playback, camera management and camera simulation. The ADP program also offers technical documentation, tools and dedicated development support for participants. See also AXIS Camera Application Platform. If you are developing solutions to support Axis network video products, take advantage of the many commercial and technical benefits available to Axis ADPs.

5. VAPIX®

All Axis network video products have an HTTP-based Application Programming Interface. VAPIX®, Axis powerful API, enables you to request images, control network camera functions (PTZ, relays,), set/retrieve internal parameter values and much more. The purpose of the API is to make it easier for developers to build applications that support Axis video products.

VAPIX® consists of:

- Axis HTTP API
- Axis Parameter Specification
- Axis RTSP API (for controlling MPEG-4 streams)

5.1. Obtaining video

For obtaining a video stream from a camera, a Motion JPEG or an MPEG-4 video stream is recommended. An alternative technique, which is not recommended, is frequent single snapshot images. In the case of single snapshots, a new HTTP connection is opened for each request.

The total number of simultaneous video streams that can be provided from an Axis video product is either 10 or 20 depending on the model. However requesting as few video streams as possible is recommended to limit the resources consumed by the camera. Additional video streams will limit the overall frame rate and increase the network bandwidth consumed by the camera.

In the case of MPEG-4, the software should only request one MPEG-4 stream, since each MPEG-4 stream cannot be configured differently from another MPEG-4 stream.

When requesting video from an Axis product with a different port configuration than the default port 80, specify the port number after the IP address or the host name of the camera.

**Example:** Using HTTP port 8080

```
http://myserver:8080/axis-cgi/mjpg/video.cgi
```

The URL for requesting MPEG-4 streams from Axis video products is:

```
<axprotocol>://<IP address>/mpeg4/<camera number>/media.amp
```

**Example:** Fetch a unicast RTP MPEG-4 stream

```
axrtpu://myserver/mpeg4/1/media.amp
```
**Example:** Fetch a multicast RTP MPEG-4 stream

```
axrtpm://myserver/mpeg4/2/media.amp
```

**Example:** Fetch an MPEG-4 stream using RTP over RTSP (RTSP port 556)

```
axrtsp://myserver:556/mpeg4/1/media.amp
```

**Example:** Fetch an MPEG-4 stream using RTP over RTSP over HTTP (HTTP port 8080)

```
axrtsphttp://myserver:8080/mpeg4/1/media.amp
```

**Note:** When requesting only video from audio capable MPEG-4 products, "/trackID=1" must be added to the URL to limit the request to video:

```
<axprotocol>://<IP address>/mpeg4/<camera number>/media.amp/trackID=1
```

### 6. Parameter configuration

Most features in Axis video products can be configured by parameters. These parameters are usually located in different configuration files in the flash memory. Since the flash memory only has limited lifetime, it is not advisable to write parameters frequently. The manufacturer estimates the maximum number of flash writes to about 100,000.

The parameters available for each camera model differs depending on the feature set. A parameter list can be obtained from the camera, either from the web interface or by using VAPIX®. Descriptions and valid values for the parameters are specified in VAPIX®, in the Parameter Specification.

The parameters are also available in the Plain Config page of the web interface - **Setup->System Options->Advanced->Plain Config**. The plain config page allows direct access to all the configurable parameters supported by the product and the page uses no scripts - Java script or any other script, and should work correctly in any browser or PDA.

Select a parameter group and all parameters in that group will be listed, along with the possibility for direct configuration.

### 7. Dynamic parameters

Axis video products support dynamic parameters, meaning that parameters can be created in runtime. Event servers, events, event actions, video motion detection windows and guard tours are examples of dynamic parameter groups. Template files resident in the video product contain a list of the parameters included in each dynamic group and the default value for each parameter.

A dynamic parameter look like this - **Event.E0.Enabled=yes**

- The first part is the parent group, in this example, **Event**.
- The E in the second part refers to the first letter of the parent group and the number thereafter is added dynamically by the system, in this example, **E0**.
- The third part is the actual parameter name, in this example, **Enabled**.
Some parameter groups can also have dynamic sub groups. Event actions are examples of a dynamic sub group to events – \texttt{Event.E0.Actions.A0.Protocol=TCP}

- The third part of the parameter, \texttt{Actions}, is the parent group to a number of parameters but also a subgroup to \texttt{Event.E0}.
- In the fourth part, \texttt{A0}, the \texttt{A} refers to the parent group \texttt{Actions} and the number is dynamically added by the system.
- The fifth part is the name of the actual parameter.

Supported dynamic parameter groups differ between the camera models depending on the feature set. The release notes file, which is available together with the latest firmware for each product, contains information about supported dynamic parameter groups and template files. \texttt{VAPIX®} contains parameter descriptions, applicable templates, and valid values for each dynamic parameter group.

8. Video motion detection

Using the built-in intelligence in the camera such as video motion detection, saves bandwidth and storage since video feed can be setup to only be transmitted when the camera detects movement in the scene.

Include windows where motion would be registered and exclude windows where motion would be discarded can be created. However, the number of video motion detection windows (include and exclude) created should be limited as much as possible since this will decrease the performance of the video product. Windows added in the same position, with the same settings should be eliminated.

Video motion detection windows are set up using dynamic parameters, examples are provided in \texttt{VAPIX®}.

9. Notification of camera events

Different kind of notifications can be sent from the camera, such as TCP, email and HTTP notifications.

Another way to retrieve information about different states in the video product is to enable trigger data in the JPEG or the GOV header of the images stream.

\texttt{VAPIX®} also contains requests for monitoring video motion detection levels, IO levels and retrieving the video status of analog camera(s) connected to a video server.

The server application can be notified in several ways. The two most common are notification via “trigger data” and creating a TCP event.

9.1. TCP notifications

TCP event requires other ports open than port 80. This can require additional configuration on the application server’s network if interfacing with remote cameras.
9.2. **HTTP upload and notifications**

The camera can be set to upload JPEG images or send notifications to an HTTP server. This setup requires an HTTP server with a receiving application that takes care of the incoming images and/or the HTTP notification sent from the camera.

9.3. **Trigger data in the video stream**

Trigger data is embedded in the camera’s video stream. A video stream from the camera is required which will always use a small amount of bandwidth.

Trigger data describes different conditions in the Axis video product, for example digital input states, video motion detection and video loss. This following text describes how to include trigger data in two different streaming formats – Motion JPEG and MPEG-4.

**Trigger data block**

The trigger data block contains the states of all triggers of interest, as opposed to including just a changed state in one block when it occurs. This allows the receiving application to lose blocks without necessarily losing a changed state.

The trigger data block format is the same for both Motion JPEG and MPEG-4. It is basically a block of text containing trigger states in this format:

```
<trigger>:<state>;<trigger>:<state>;...
```
<trigger> is a tag for the trigger and <state> is a text describing the state. The following table lists defined trigger tags and their possible states.

<table>
<thead>
<tr>
<th>Trigger tag</th>
<th>Description</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 … An</td>
<td>Status for audio trigger for source 0 to n, where n is the total number of audio sources minus one.</td>
<td>1 0</td>
</tr>
<tr>
<td>IO0 … IOn</td>
<td>Status for digital input 0 to n, where n is the total number of inputs minus one. Note that state &quot;1&quot; means that the input has triggered, which isn't necessarily the same as that the input is high. Each input can be configured when to trigger.</td>
<td>0 1</td>
</tr>
<tr>
<td>V0 ... Vn</td>
<td>Video loss status for video source 0 to n, where n is the total number of video sources minus one. State &quot;0&quot; means that there is no video for that video source.</td>
<td>0 1</td>
</tr>
<tr>
<td>M0 ... Mn</td>
<td>Video motion detection status for window 0 to n, where n is the total number of video motion detection windows minus one. State &quot;1&quot; means that motion has been detected for this window, that is, the motion level is above the configured threshold.</td>
<td>0 1</td>
</tr>
<tr>
<td>ML0 ... MLn</td>
<td>Video motion detection level for window 0 to n, where n is the total number of video motion detection windows minus one.</td>
<td>000 … 100</td>
</tr>
<tr>
<td>T0 … Tn</td>
<td>Camera tampering status for source 0 to n, where n is the total number of video sources minus one. State &quot;1&quot; means that camera tampering has been detected for that video source.</td>
<td>0 1</td>
</tr>
</tbody>
</table>

New trigger tags may be added in the future. Any trigger tag not understood by the interpreter should simply be ignored.

**Example:**

```
A0:0;IO0:0;IO1:1;IO2:1;IO3:0;V0:0;M0:1;ML0:035;T0:1;
```

This should be interpreted as that audio is not triggered, input 0 and 3 are not triggered and that input 1 and 2 are triggered. Video source 0 has video. Motion is detected for window 0. Motion level for window 0 is 35 and camera tampering is triggered on video source 0.
Motion JPEG

For Motion JPEG the trigger data block is included as a comment header for every image. Each image may contain several comments in this format:

<table>
<thead>
<tr>
<th>Field</th>
<th>Size (bytes)</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>marker</td>
<td>2</td>
<td>0xFF 0xFE</td>
<td>JPEG comment marker</td>
</tr>
<tr>
<td>length</td>
<td>2</td>
<td>4...260</td>
<td>Length of the comment</td>
</tr>
<tr>
<td>axis_id</td>
<td>1</td>
<td>0x0A</td>
<td>This marks the comment as defined by Axis</td>
</tr>
<tr>
<td>trigger_id</td>
<td>1</td>
<td>0x03</td>
<td>This marks the comment as trigger data</td>
</tr>
<tr>
<td>data</td>
<td>length-4</td>
<td></td>
<td>The payload = the trigger data block</td>
</tr>
</tbody>
</table>

MPEG-4

For MPEG-4 the trigger data block is included as "user data" in the GOV header. The GOV header may contain several data blocks following each other in this format:

<table>
<thead>
<tr>
<th>Field</th>
<th>Size (bytes)</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>marker</td>
<td>4</td>
<td>0x00 0x00 0x01 0xB2</td>
<td>User data start code</td>
</tr>
<tr>
<td>axis_id</td>
<td>1</td>
<td>0x0A</td>
<td>This marks the user data as defined by Axis</td>
</tr>
<tr>
<td>trigger_id</td>
<td>1</td>
<td>0x03</td>
<td>This marks the user data as trigger data</td>
</tr>
<tr>
<td>data</td>
<td>Length of trigger data</td>
<td>1 ... 256</td>
<td>The payload = the trigger data block</td>
</tr>
</tbody>
</table>

The GOV header is inserted into the MPEG-4 stream at regular intervals, typically every 8th frame (this is configurable). A new GOV header can also be forced to be inserted as soon as possible when a trigger changes state. This is done for I/O triggers to reduce the latency.

Parameter Configuration

The following parameters control inclusion of trigger data in the video stream:

**Note:** The product must be restarted for the configuration to take effect.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Default values</th>
<th>Valid values</th>
<th>Security level (get/set)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image. TriggerDataEnabled</td>
<td>no</td>
<td>no, yes</td>
<td>4/4</td>
<td>This parameter enables or disables the triggered daemon, which is responsible for including trigger data in the video</td>
</tr>
</tbody>
</table>
stream. You must set it to "yes" in order to make the other parameters work.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image.I#.MPEG. UserDataEnabled</td>
<td>no</td>
<td>no, yes</td>
<td>Enable/disable insertion of user data in the MPEG GOV header. Note that this parameter was introduced and set to no by default in firmware versions higher than 4.20. The user data was included by default in earlier firmware versions.</td>
</tr>
<tr>
<td>Image.I#.TriggerData. AudioEnabled</td>
<td>yes</td>
<td>no, yes</td>
<td>This parameter enables inclusion of audio trigger states for image configuration #. Setting it to &quot;yes&quot; will include the states of all audio triggers in the trigger data for this image configuration. If you just want to include some audio triggers you should set it to &quot;no&quot; and use the UserTriggers parameter instead.</td>
</tr>
<tr>
<td>Image.I#.TriggerData. IOEnabled</td>
<td>yes</td>
<td>no, yes</td>
<td>This parameter enables inclusion of digital input states for image configuration #. Setting it to &quot;yes&quot; will include the states of all digital inputs in the trigger data for this image configuration. If you just want to include some digital inputs you should set it to &quot;no&quot; and use the UserTriggers parameter instead.</td>
</tr>
<tr>
<td>Image.I#.TriggerData. MotionDetection Enabled</td>
<td>yes</td>
<td>no, yes</td>
<td>This parameter enables inclusion of video motion detection states for image configuration #. Setting it to &quot;yes&quot; will include the video motion detection states of the windows that belong to this image source in the trigger data. If you just want to include some window states or window states belonging to other image sources you should set it to &quot;no&quot; and use the UserTriggers parameter instead.</td>
</tr>
<tr>
<td>Image.I#.TriggerData. MotionLevelEnabled</td>
<td>no</td>
<td>no, yes</td>
<td>This parameter enables inclusion of video motion detection levels for image configuration #. Setting it to &quot;yes&quot; will include the video motion detection levels of the windows that belong to this image source in the trigger data. If you just want to include some window levels or window levels belonging to other image sources you should set it to &quot;no&quot; and use the UserTriggers parameter instead.</td>
</tr>
<tr>
<td>Image.I#.TriggerData.</td>
<td>yes</td>
<td>no, yes</td>
<td>This parameter enables inclusion of camera tampering trigger states for image configuration #. Setting it to &quot;yes&quot; will include the camera tampering trigger states of the windows that belong to this image source in the trigger data. If you just want to include some camera tampering trigger states or camera tampering trigger states belonging to other image sources you should set it to &quot;no&quot; and use the UserTriggers parameter instead.</td>
</tr>
</tbody>
</table>
### TamperingEnabled

Setting it to "yes" will include the camera tampering state of the image configuration in the trigger data. If you want to include camera tampering states for other image configurations you should use the UserTriggers parameter.

<table>
<thead>
<tr>
<th>Image.I# TriggerData. VideoLossEnabled</th>
<th>no, yes</th>
<th>4/4</th>
</tr>
</thead>
</table>

This parameter enables inclusion of video loss status for image configuration #. Setting it to "yes" will include the video loss status of the image source in the trigger data. If you want to include video loss states for other image sources you should use the UserTriggers parameter.

<table>
<thead>
<tr>
<th>Image.I# TriggerData. UserTriggers</th>
<th>A string</th>
<th>4/4</th>
</tr>
</thead>
</table>

This string enables inclusion of user defined triggers for image configuration #. The string should be in the format "<trigger1>;<trigger2>;<trigger3>;...". See example 2 below.

In the following examples we have a video server with four video inputs and four digital inputs. We have configured one video motion detection window per video input.

#### Example 1: Default configuration

Set Image.TriggerDataEnabled="yes" to enable trigger data. Apart from that we use the default configuration:

```plaintext
Image.I0.TriggerData.IOEnabled="yes"
Image.I0.TriggerData.AudioEnabled="yes"
Image.I0.TriggerData.MotionDetectionEnabled="yes"
Image.I0.TriggerData.MotionLevelEnabled="no"
Image.I0.TriggerData.TamperingEnabled="yes"
Image.I0.TriggerData.VideoLossEnabled="yes"
Image.I0.TriggerData.UserTriggers=""
Image.I1.TriggerData.AudioEnabled="yes"
Image.I1.TriggerData.IOEnabled="yes"
Image.I1.TriggerData.MotionDetectionEnabled="yes"
Image.I1.TriggerData.MotionLevelEnabled="no"
Image.I1.TriggerData.TamperingEnabled="yes"
Image.I1.TriggerData.VideoLossEnabled="yes"
Image.I1.TriggerData.UserTriggers=""
Image.I2.TriggerData.AudioEnabled="yes"
Image.I2.TriggerData.IOEnabled="yes"
Image.I2.TriggerData.MotionDetectionEnabled="yes"
Image.I2.TriggerData.MotionLevelEnabled="no"
Image.I2.TriggerData.TamperingEnabled="yes"
Image.I2.TriggerData.VideoLossEnabled="yes"
Image.I2.TriggerData.UserTriggers=""
Image.I3.TriggerData.AudioEnabled="yes"
Image.I3.TriggerData.IOEnabled="yes"
Image.I3.TriggerData.MotionDetectionEnabled="yes"
Image.I3.TriggerData.MotionLevelEnabled="no"
```
For image configuration 0 this would result in a trigger data block similar to this:

A0:0;IO0:0;IO1:0;IO2:0;IO3:0;M0:0;T0:0;V0:0;

And for image configuration 1:

A0:0;IO0:0;IO1:0;IO2:0;IO3:0;M1:0;T1:0;V1:0;

And for image configuration 2:

A0:0;IO0:0;IO1:0;IO2:0;IO3:0;M2:0;T2:0;V2:0;

And for image configuration 3:

A0:0;IO0:0;IO1:0;IO2:0;IO3:0;M3:0;T3:0;V3:0;

Example 2: All in one

Include all trigger states in the trigger data block for image configuration 0:

Image.I0.TriggerData.IOEnabled="yes"
Image.I0.TriggerData.AudioEnabled="yes"
Image.I0.TriggerData.MotionDetectionEnabled="no"
Image.I0.TriggerData.MotionLevelEnabled="no"
Image.I0.TriggerData.TamperingEnabled="no"
Image.I0.TriggerData.VideoLossEnabled="no"
Image.I0.TriggerData.UserTriggers="M0;M1;M2;M3;ML0;ML1;ML2;ML3;T0;T1;T2;T3;V0;V1;V2;V3"

This would result in a trigger data block similar to this:

A0:0;IO0:0;IO1:0;IO2:0;IO3:0;M0:0;M1:0;M2:0;M3:0;ML0:000;ML1:000;ML2:000;ML3:000;T0:0; T1:0;T2:0;T3:0;V0:0;V1:0;V2:0;V3:0;

9.4. Dynamic event notification

From firmware version 4.30 it is possible to use specifiers together with plain text, to be able to send dynamic event notifications. The specifiers can be used in these cases:

- E-mail upload/notification, the parameters Event.E#.Actions.A#.Subject and Event.E#.Actions.A#.Message
- HTTP notification, the parameter Event.E#.Actions.A#.Message
- TCP notification, the parameter Event.E#.Actions.A#.Message

The valid format codes are as follows:

- #e = The event name.
• #E = The event description. Default texts are defined in the file /etc/event_desc.list located in the product’s internal file system. The texts can be modified using the built-in editor:
  http://myserver/admin-bin/editcgi.cgi?file=/etc/event_desc.list.

• The texts are defined per event and constitute the following:
  “Video loss on video %d” where %d is the video channel number starting from 1.
  “Restarting”
  “Manual trigger input port(s) %s went %s” where the first %s can be a number starting from 5 and the second %s can be active or inactive.
  “Motion ‘%s’ on motion window ‘%s’” where the first %s can be start or stop and the second %s is the window name.
  Input port(s) %s went %s” where the first %s is the number of the input port starting from 1 and the second %s can be active or inactive.

• #w = The video motion detection window name.

• #t = The trigger type and value.

Examples:

“V#:\” when video is lost, where # is the number of the video channel.

“BOOT:/” when the product starts.

“I0#:\” or “I0#/” when an input port or a virtual input is triggered, where # is the number of the input port or the virtual input port.

“M#:\” or “M#/” for video motion detection where # is the number of the motion window.
Also other specifiers can be used:

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Description</th>
<th>Syntax/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>%c</td>
<td>The date and time.</td>
<td>Example: Wed Sep 19 23:15:33 2007</td>
</tr>
<tr>
<td>%D</td>
<td>The date.</td>
<td>MM/DD/YY Example: 12/31/07</td>
</tr>
<tr>
<td>%F</td>
<td>The date.</td>
<td>YYYY-MM-DD Example: 2007-12-31</td>
</tr>
<tr>
<td>%x</td>
<td>The date representation without the time.</td>
<td>MM/DD/YY Example: 09/19/07</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>%p</td>
<td>AM or PM, according to the given time value or the corresponding strings for the current locale. Noon is treated as PM and midnight as AM.</td>
<td>Example: PM</td>
</tr>
<tr>
<td>%r</td>
<td>The time in a.m. or p.m. notation.</td>
<td>HH:MM:SS AM Example: 05:37:17 PM</td>
</tr>
<tr>
<td>%R</td>
<td>The time in 24-hour notation.</td>
<td>HH:MM Example: 17:37</td>
</tr>
<tr>
<td>%T</td>
<td>The time in 24-hour notation.</td>
<td>HH:MM:SS Example: 17:41:10</td>
</tr>
<tr>
<td>%X</td>
<td>The time representation without the date.</td>
<td>HH:MM:SS Example: 17:41:10</td>
</tr>
<tr>
<td>%z</td>
<td>The time zone as hour offset from GMT. Required to emit RFC822-conformant dates (using &quot;%a, %d %b %Y %H:%M:%S %z&quot;).</td>
<td>Example: +000</td>
</tr>
<tr>
<td>%Z</td>
<td>The time zone or name or abbreviation.</td>
<td>Example: UTC</td>
</tr>
<tr>
<td><strong>Year</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>%C</td>
<td>The century number (year/100) as a 2-digit number.</td>
<td>Example: 20</td>
</tr>
<tr>
<td>%g</td>
<td>As for %G, but without the century, that is, with a 2-digit year (00-99).</td>
<td>Example: 38</td>
</tr>
<tr>
<td>%G</td>
<td>The ISO 8601 year with century as a decimal number. The 4-digit year corresponding to the</td>
<td>Example: 38</td>
</tr>
</tbody>
</table>
ISO week number (see %V). This has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, then that year is used instead.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>%y</td>
<td>The year as a 2-digit number without the century (range 00 to 99).</td>
<td>07</td>
</tr>
<tr>
<td>%Y</td>
<td>The year as a decimal number, including the century.</td>
<td>2007</td>
</tr>
</tbody>
</table>

**Month**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>%b</td>
<td>The abbreviated month name.</td>
<td>Sep</td>
</tr>
<tr>
<td>%B</td>
<td>The full month name.</td>
<td>September</td>
</tr>
<tr>
<td>%h</td>
<td>Equivalent to %b.</td>
<td>Sep</td>
</tr>
<tr>
<td>%m</td>
<td>The month as a 2-digit number (range 01 to 12).</td>
<td>09</td>
</tr>
</tbody>
</table>

**Week**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>%U</td>
<td>The week number of the current year as a 2-digit number, range 00 to 53 starting with the first Sunday as the first day of week 01. See also %V and %W.</td>
<td>37</td>
</tr>
<tr>
<td>%V</td>
<td>The ISO 8601:1988 week number of the current year as a 2-digit number, in the range 01 to 53, where week 1 is the first week that has at least 4 days in the current year and with Monday as the first day of the week. See also %U and %W.</td>
<td>38</td>
</tr>
<tr>
<td>%W</td>
<td>The week number of the current year as a 2-digit number, in the range 00 to 53 starting with the first Monday as the first day of week 01.</td>
<td>38</td>
</tr>
</tbody>
</table>

**Day**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>%a</td>
<td>The abbreviated weekday name.</td>
<td>Wed</td>
</tr>
<tr>
<td>%A</td>
<td>The full weekday name.</td>
<td>Wednesday</td>
</tr>
<tr>
<td>%d</td>
<td>The day of the month as a 2-digit number (range 01 to 31).</td>
<td>01</td>
</tr>
<tr>
<td>%e</td>
<td>Day of the month (range 1-31). As for %d, but a leading zero is replaced by a blank space.</td>
<td>1</td>
</tr>
</tbody>
</table>
%j    The day of the year as a decimal number (range 001 to 366).    Example: 054

%u    The day of the week as a number, range 1 to 7, Monday = 1. See also %w.    Example: 1

%w    The day of the week as a number, in the range 0 to 6. Sunday = 0. See also %u.    Example: 1

### Hour

%H    The hour as a 2-digit number, using the 24-hour clock (range 00 to 23).    Example: 17

%I    The hour as a 2-digit number, using the 12-hour clock (range 01 to 12).    Example: 05

%k    The hour (24-hour clock) as a number (range 0 to 23); single digits are preceded by a blank space. (See also %H.)    Example: 9

%l    The hour (12-hour clock) as a number (range 1 to 12); single digits are preceded by a blank. (See also %I.)    Example: 5

### Minute

%M    The minute as a decimal number (range 00 to 59).    Example: 37

### Second

%f    The 1/100 second fraction.    Example: 87

%s    The number of seconds since the Epoch, that is, since 1970-01-01 00:00:00 UTC.    Example: 1189599826

%S    The second as a 2-digit number (range 00 to 60).    Example: 09

### Product/system information

#i    The IP address.    Example: 10.13.24.88

#m    The short MAC address (last 6 characters).    Example: 77:F8:26

#M    The long MAC address (all characters).    Example: 00:40:8C:77:F8:26

#n    The host name.    Example: Axis-00408c77f826
### Video information

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>#b</td>
<td>The bit rate in Kbit/s with no decimals.</td>
<td>2661</td>
</tr>
<tr>
<td>#B</td>
<td>The bit rate in Mbit/s with two decimals.</td>
<td>2.66</td>
</tr>
<tr>
<td>#r</td>
<td>The frame rate with two decimals.</td>
<td>25.01</td>
</tr>
<tr>
<td>#R</td>
<td>The frame rate (no decimals).</td>
<td>25</td>
</tr>
<tr>
<td>#s</td>
<td>The sequence number with 5 digits.</td>
<td>00345</td>
</tr>
<tr>
<td>#v</td>
<td>The video source number.</td>
<td>2</td>
</tr>
<tr>
<td>#w</td>
<td>The name of the motion detection window triggering an event</td>
<td>GarageDoor</td>
</tr>
</tbody>
</table>

### Pan/Tilt/Zoom information

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>#L</td>
<td>PTZ locator zone name – OSDI zone</td>
<td>ParkingLot</td>
</tr>
<tr>
<td>#p</td>
<td>Preset number – if at a preset position, otherwise blank.</td>
<td>3</td>
</tr>
<tr>
<td>#P</td>
<td>Preset name – if at a preset position, otherwise blank.</td>
<td>Garage</td>
</tr>
<tr>
<td>#x</td>
<td>Pan coordinate (signed, with two decimals).</td>
<td>-77.61</td>
</tr>
<tr>
<td>#y</td>
<td>Tilt coordinate (signed, with two decimals).</td>
<td>-7.61</td>
</tr>
<tr>
<td>#z</td>
<td>Zoom coordinate (1 to 19999).</td>
<td>15444</td>
</tr>
<tr>
<td>#Z</td>
<td>Zoom magnification (one decimal place).</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Other information

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%n</td>
<td>The new line character.</td>
</tr>
<tr>
<td>%t</td>
<td>The tab character.</td>
</tr>
<tr>
<td>%%</td>
<td>The % character.</td>
</tr>
<tr>
<td>##</td>
<td>The # character.</td>
</tr>
</tbody>
</table>

Note that supported specifiers are product and firmware dependent.


10. PTZ configuration and control

The PTZ preset position parameters may vary between the Axis camera and video server models. Available positions can however be retrieved the same way from all PTZ enabled products using VAPIX®, which makes it possible to use the same implementation for all Axis products.

**Examples**

Request available PTZ preset positions:

```
http://myserver/axis-cgi/com/ptz.cgi?query=presetposall
```

**Response:**

```
Preset Positions for camera 1
presetposno0=Default
presetposno1=Door
presetposno2=Window
```

Go to a PTZ preset position:

```
http://myserver/axis-cgi/com/ptz.cgi?gotoserverpresetname=Door
```

11. Privacy masking and overlay images

Most Axis video products support privacy masking in images, that is, configurable areas of solid color can be used to block out certain parts of the video image. The privacy masks are shown as solid areas of color in the image. They are always located in the configured area and are not repositioned while using text overlay. Privacy masks cannot be excluded from the image while requesting an image using VAPIX® and it is not possible to request a rotated image when the image contains privacy mask areas.

Overlay images can be used for showing, for example, a logotype in the video stream. Although an overlay image can also conceal parts of the image, the overlay can be bypassed when the video image is accessed using VAPIX®.

12. Supported image resolutions

Axis video products support more resolutions using VAPIX® than can be seen in the product's web interface. Available resolutions for each product can be found in the document Supported HTTP API Image Resolutions and in the product's firmware release notes. The parameter Properties.Image.Resolution contains the same resolutions that can be found in the product's web interface.

13. Properties parameters

Axis video products have Properties parameters that contain information about product capability. These parameters can be useful from an application point of view to check for the functionalities
supported by a specific product. For example, does the product support image formats, audio
formats, video motion detection and PTZ etc.

Example: The Properties parameters in an AXIS 233D

```plaintext
root.Properties.API.HTTP.Version=2
root.Properties.API.HTTP.AdminPath=/operator/basic.shtml
root.Properties.Audio.Audio=yes
root.Properties.Audio.Format=g711,g726
root.Properties.Firmware.BuildNumber=8
root.Properties.GuardTour.GuardTour=yes
root.Properties.HTTPS.HTTPS=yes
root.Properties.Image.Rotation=0,180
root.Properties.Image.Resolution=4CIF,2CIFEXP,2CIF,CIF,QCIF
root.Properties.Image.Format=jpeg,mjpeg,mpeg4
root.Properties.Motion.Motion=yes
root.Properties.PTZ.PTZ=yes
root.Properties.PTZ.DigitalPTZ=no
root.Properties.System.Language=English
root.Properties.System.SerialNumber=00408C815F6D
```

14. Developer information

The firmware release notes document for each product has an application Developer Information
section that contains information about the sections of VAPIX® supported, available template files
for dynamic parameters and supported image resolutions.

15. Axis Windows SDKs and development tools

AXIS Media Control SDK is publicly available at www.axis.com; however, Axis offers more
Windows SDKs and tools within the ADP program. Below you can find descriptions of the SDKs
to use for different applications.

15.1. AXIS Media Control SDK

AXIS Media Control SDK enables extremely easy integration of viewing Motion JPEG and
MPEG-4 streams directly in Microsoft Internet Explorer, Visual Basic and other ActiveX
containers. Key features for AXIS Media Control SDK include:

- Motion JPEG support
- MPEG-4 support
- Audio support
- Functionality to record and playback ASF files
- PTZ control using mouse or joystick

For many applications AXIS Media Control is enough. This is a good development tool and easy
to use for viewing live images and recording images such as when an event occurs, for example.
15.2. AXIS Media Parser SDK

The AXIS Media Parser SDK contains two components, AXIS Media Parser and AXIS Media Viewer. The AMP SDK is designed for performance, scalability and flexibility, and it enables easy integration of live view, and recording and playback using any media format supported by Axis network video products. Key features for the AXIS Media Parser SDK include:

- Motion JPEG support
- MPEG-4 support
- Synchronized audio support
- Support for third party DirectShow filters
- Functionality to write media to ASF files

Integrating support for advanced recording and playback can be very complex. AXIS Media Parser SDK is a COM-object that hides the complexity of DirectShow filters and parsing of MPEG-4 video streams. It offers functionality such as recording of all types of media formats as well as advanced playback features like fast forward, rewind and stepping in the stream. It is a very suitable tool to use for video and audio recording, playback of recorded media, extracting trigger data from the video stream, and for intelligent video analysis. AXIS Media Parser is independent of a graphical user interface and can be run as a service in the background. AXIS Media Parser is exclusively available to members of the Axis ADP Program.

15.3. AXIS Camera Management SDK

AXIS Camera Management SDK contains functionality to discover and manage Axis video devices on a network. The component is used by the AXIS Camera Management software application for all network-related functionality and can be used by any application. It supports discovery, configuration, status and upgrade of Axis video devices. AXIS Camera Management SDK is exclusively available to members of the Axis ADP Program.

15.4. AXIS Joystick SDK

AXIS Joystick SDK is a COM component that provides an easy-to-use programming interface for adding joystick support to your application. It enables PTZ control for all Axis pan/tilt/zoom cameras, dome cameras and video encoders. This SDK makes it very easy to manage and configure connected joysticks, as well as receive events when a button is pressed or a joystick moved. Use this component instead of the joystick support in AXIS Media Control SDK if you want full control over connected joysticks. AXIS Joystick SDK is exclusively available for members of the Axis ADP Program.

15.5. AXIS Virtual Camera

AXIS Virtual Camera enables you to scale up your application tests by simulating a large number of real cameras. Replace a Motion JPEG or MPEG-4 stream with one that is previously recorded or replace generic streams or web pages creating the appearance that they come from a true device. Key features of AXIS Virtual Camera include:

- Motion JPEG support
- MPEG-4 (multicast and unicast) support
- Synchronized audio support
- Support for virtual parameters
• Parameter templates for easier configuration

AXIS Virtual Camera can simulate more than 100 cameras in full resolution on a standard Windows PC (Intel Pentium 4, 3.0GHz). This tool is exclusively available for members of the Axis ADP Program.

15.6. AXIS Custom Firmware Tool

AXIS Custom Firmware Tool can be used to create your own firmware from a preconfigured Axis video product. Configure your product (create new users, add events, etc.), add your own scripts, web pages, C-applications and so forth. When ready, create a custom firmware that can be used to upgrade other Axis video products of the same model. When setting the products upgraded with the custom firmware to factory default, these products will all have the same configuration, scripts etc. that the product the firmware was created from. AXIS Camera Management can preferably be used for upgrading many video products with the custom firmware and setting them to factory default. AXIS Custom Firmware Tool is exclusively available for members of the Axis ADP Program.

15.7. Distribution

There are several choices available for redistributing Axis components, and these choices are described in detail in the SDK documentation. Decoders are subject to specific licensing conditions and are not distributed with the SDKs. More information can be found here - http://www.axis.com/techsup/software/amc/license.htm.

16. Hardware and supported features

The hardware used in Axis video products differs between models. Specifications can be found in the product's data sheet. A cheat sheet of the hardware used in each product is provided in the Product Interface Guide. Product comparison guides are also available for network cameras and video encoders, which will give you an overview of supported features in each video product model. The Properties parameters are also useful from an application point of view. See section on Properties Parameters for more information.

17. Embedded development

Axis open systems allow developers to add custom scripts and applications to be run directly in the video products, which makes it possible to tailor the functionality to specific needs.

17.1. Considerations

Remember that if you want to add your own scripts, applications, or web pages, the memory resources available in Axis video products are limited. The available memory resources are different for different products, and they can even vary between products of the same model depending on the configuration. Check the product's data sheet or the Product Interface Guide to see how much memory the product has.

The amount of writing to the flash chip is limited - estimated to be about 100,000 times by the manufacturer. If you have temporary data save it in the RAM memory (/tmp) instead of using the flash.

When upgrading firmware, custom scripts and applications are erased. Make sure to have a backup.
17.2. Shell scripting and custom web interface

Axis video products allow you to use embedded shell scripting and to add your own custom web pages to the product to tailor the functionality to your needs. Please read Axis Scripting Guide before getting started. Support for development of shell scripts and custom web pages are exclusively available for Axis ADP partners.

17.3. Application development using C

SDKs for developing C-applications to be run on top of the firmware in Axis video products are available for Axis ADP partners. Embedded Development SDKs are available for a selection of Axis video products, and are provided as a tool for proficient Linux programmers. Axis does not provide additional technical support for embedded development.

18. Axis tools

Axis tools enable easy installation and can calculate storage and bandwidth. Learning about image quality, best installation, and best set up of the camera gives you much better conditions for image analysis in your application.

18.1. AXIS Camera Management

AXIS Camera Management is a powerful and efficient installation and management tool especially designed to be used with Axis network video products. Using standard network protocols and methods, the tool can automatically find and set IP addresses and configure devices, and indicate the connection status and manage firmware upgrades of multiple devices.

18.2. AXIS Installation Monitor

AXIS Installation Monitor software simplifies the physical installation and service of Axis network cameras mounted on walls, ceilings and poles -- wherever focus and direction need to be set. Basic camera properties can be monitored from a handheld Pocket PC, without having to rely on a laptop or contact with a colleague at a remote computer.

AXIS Installation Monitor automatically discovers Axis network video products on the network. The installer can view the live camera images by using a Pocket PC while adjusting the focus, viewing angle and direction and even the field of view for pan/tilt cameras. The built-in focus indicator makes the installation quick and efficient.

18.3. AXIS Design Tool

The AXIS Design Tool is a simulation-based calculation tool that allows you to determine bandwidth and storage requirements for specific network video projects by guiding you through camera selection and frame rate settings for viewing and recording. Information can be saved for later use and exported to a file, enabling you to share the data with other tool users.

18.4. AXIS Image Quality Tool

AXIS Image Quality Tool helps you understand how conditions such as light, scene and camera settings affect the final image.
18.5. **AXIS Lens Calculator Tool**

AXIS Lens Calculator Tool helps you calculate the focal length of the lens you will need in order to capture a specific scene at a certain distance.

18.6. **AXIS Viewfinder**

AXIS Viewfinder is a complement to the lens calculator. It helps you choose the correct camera and lens combination for your specific application by comparing different angles of view. The two determining components affecting the angle of view are the camera sensor and the focal length of the lens.

19. **Support**

Axis offers support for product related questions at [www.axis.com](http://www.axis.com) where you can find an FAQ database and an online helpdesk. Development support is, however, exclusively available for members of the Axis ADP program.

AXIS Forum is a community for customers, partners, developers and other interested parties to exchange ideas and experiences around Axis products.

20. **References**

Axis' Application Development Partner Program:


VAPIX®, including HTTP API, RTSP API and Parameter Specification:


Windows development:


Embedded scripting:


Product Interface Guide: