SPINETIX

SpinetiX Technical Documentation

Streaming video v2.1

Revision: 2010, March 3

DISCLAIMER

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS. THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT IS SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR SPINETIX REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. SPINETIX DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

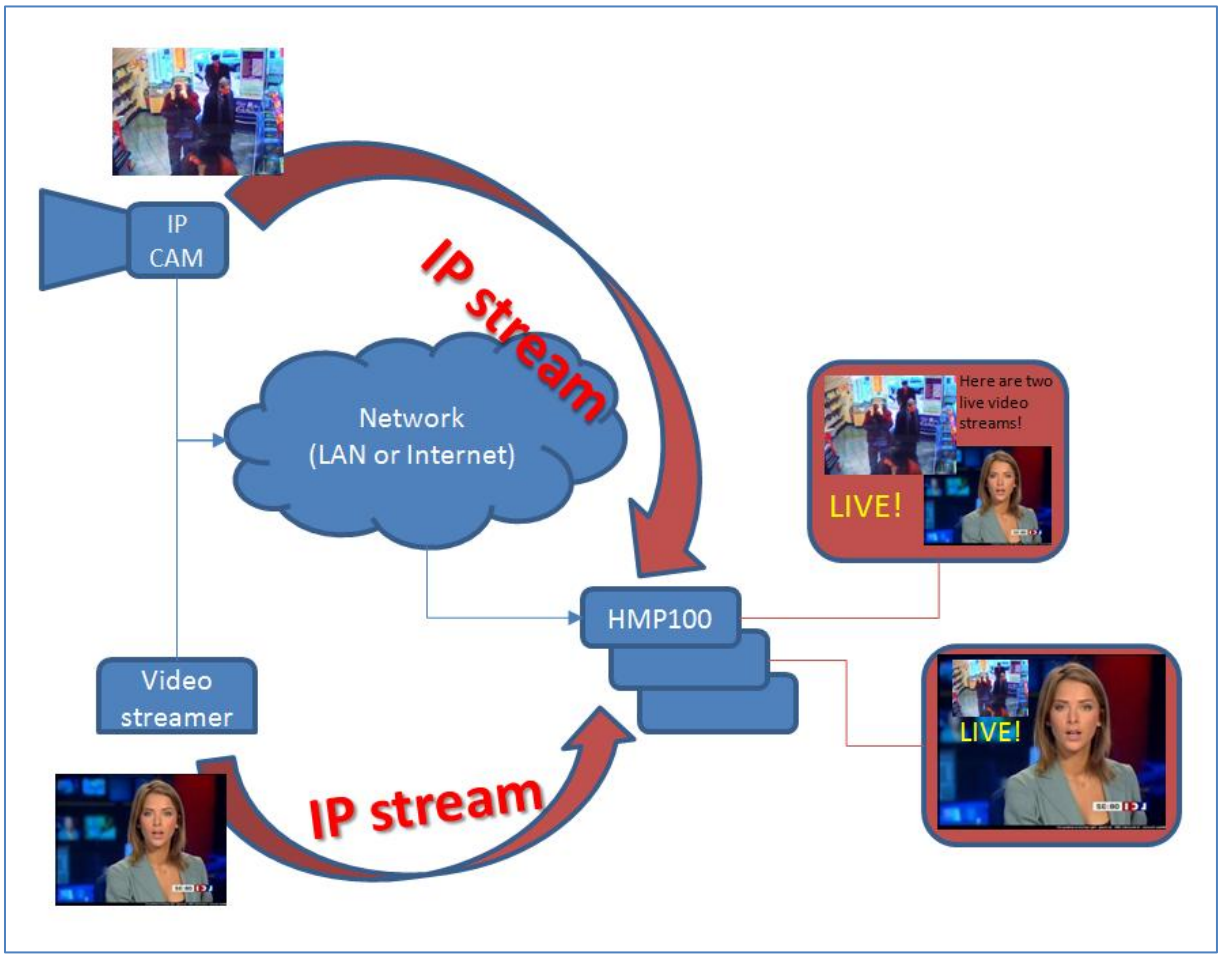
IN NO EVENT SHALL SPINETIX OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF SPINETIX OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Hyper Media Player HMP100 User Manual v2.0

SpinetiX S.A.
Rue des Terreaux 17
1003 Lausanne
Switzerland
www.spinetix.com
info@spinetix.com

INTRODUCTION

One of the most requested applications for the HMP100 is the reproduction of live video streaming. Just like a standard local or remote video, a streaming video is considered an object to be placed in a project played by the HMP100. Thus infinite possibilities of integration exist: streaming video can appear in a box with several other objects or full screen, like it would be on a traditional TV. The architecture shown below is a typical application of video streams sent through an IP network, received by an HMP100 and displayed on the screen with other contents.



This document explains and shows in detail how to configure a streaming source on a side and the HMP100 on the other side to receive and display the desired media. A few real life examples are also provided. Both hardware and software streamers are taken into consideration, so there is no need to buy an expensive device to test the streaming functionality with the HMP100.

A troubleshooting section is also available, explaining what are the most frequent causes of issues during the setup of the streaming architecture.

TABLE OF CONTENTS

INTRODUCTION TO STREAMING

Application examples

Several traditional TV channels are now also available on the Internet, so for example the TV content can be integrated with local news feeds and advertisement.

IP cameras have become very cheap and popular, the video produced by a camera could be shown by the HMP100 for security purposes together with other information provided by other devices.

IP video streamers are also very common, a small box can convert several analog videos into IP streams, this paves the way for IP TV and Video on Demand applications.

Streaming methods and protocols:

The particularity of streaming is the fact that the media is played while it is still being downloaded. This is very different from downloading a complete video and, only after, starting to play it. Thus two big advantages are provided by the streaming feature:

1. no need to wait for the whole video to be downloaded before being able to watch it;
2. possibility to stream live events (tv news, games, shows, etc).

Several protocols used to broadcast streaming audio and video exist, the most common are listed below and each of them is supported by the HMP100:

HTTP streaming is a mechanism for sending data from a Web server to a Web client without terminating the response to the client as it would happen with a typical HTTP request.

MMS uses the Microsoft Media Services protocol to transfer unicast data. MMS can be transported via UDP or TCP. If the client cannot negotiate a good connection using MMS over UDP, it will resort to MMS over TCP or HTTP over TCP. This is not as ideal for streaming as MMS over UDP, but ensures connectivity nonetheless.

RTSP (Real Time Streaming Protocol), developed by the IETF (RFC 2326), is a protocol to be used in streaming media systems allowing a client to control a streaming media server, issuing commands like "play" and "pause", and allowing on demand access to files on a server. Some RTSP servers use RTP as the transport protocol for the actual audio/video data.

STREAMING VIDEO IN HMD PROJECTS

Streaming "http" video with an "asx" file

If the video is available through a streaming server using http streaming, an ASX/WSX file is needed to connect to the source. The ASX/WSX file is generally created by the video server itself. To use the ASX file provided by the video server follow the steps below:

1. Create a multilayer document and open it;
2. Configure the properties of the main media (button "...");
3. Change the URI string to the path of the chosen ASX file; e.g.: "http://mfile.akamai.com/21529/live/reflector:22961.asx"
4. Open the "More..." tab, and make sure that the Mime type is set to "video/x-ms-asf";
5. Close the dialog and apply/save your changes.

Streaming "http" video

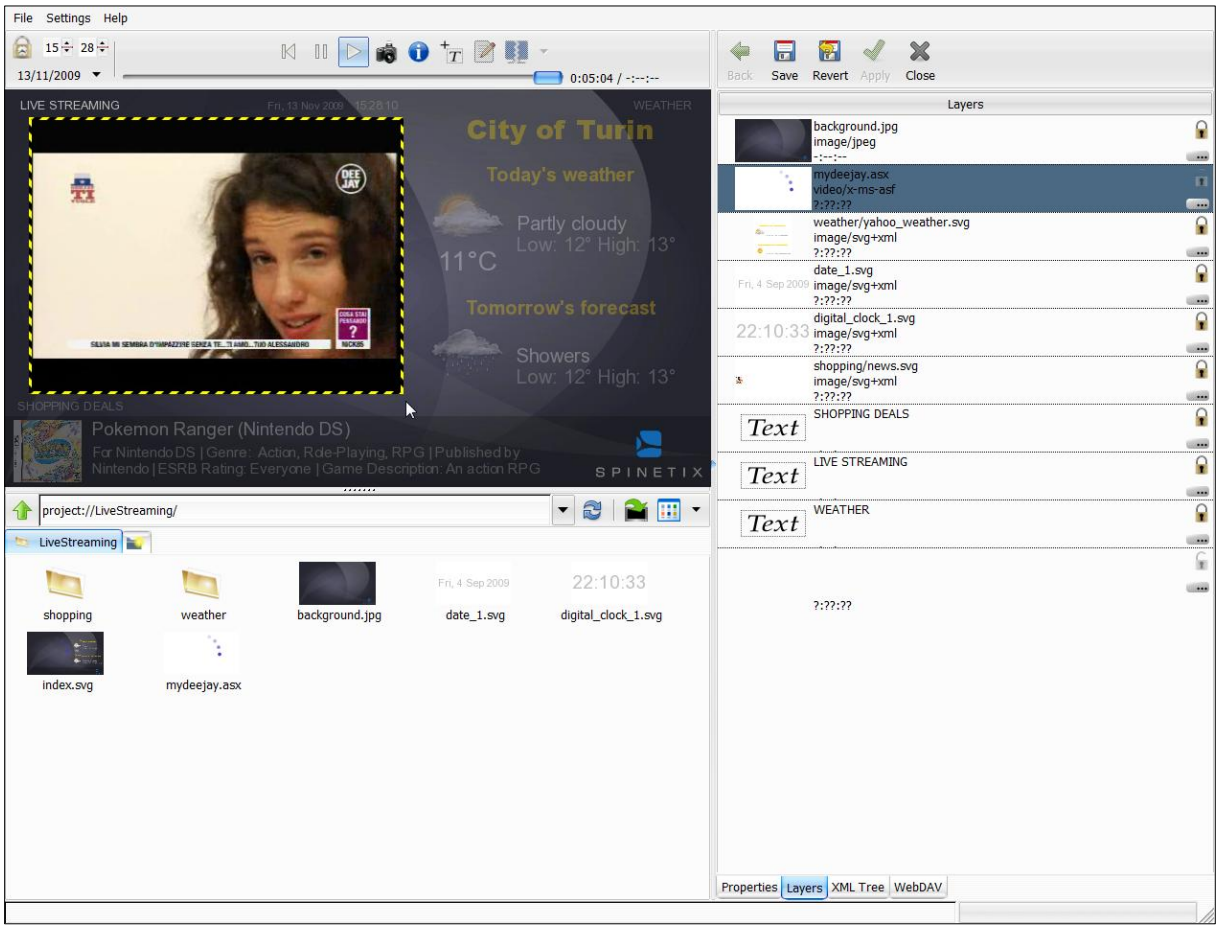
If the selected streaming server doesn't provide an ASX file, the ASX file must be created manually, with the following procedure:

1. Open a simple XML editor (or text editor, like Notepad)
2. Create the following file, replacing the HREF value by the selected media source:

```
<ASX version="3.0">
<ENTRY>
<REF HREF="http://live.mediaserver.kataweb.it/deejaytv" />
</ENTRY>
</ASX>
```

3. Save this file as "media.asx"
4. Include this file into the HMD project, as if it was a local video file.

The picture below shows a typical project where a layer (highlighted) consists in a ASX file and is managed by HMD exactly like any other layer (an image, a video, etc.):



The highlighted layer above refers to the ASX file shown below:

Streaming video using "mms" or "rtsp"

If the selected video server provides a video stream using the "mms" or the "rtsp" streaming protocol, the player can connect directly to the server. The following procedure is required:

1. Create a multilayer document and open it;
2. Configure the properties of the main media;
3. Change the URI string to the path of the selected server. For instance:
   mms://live.france24.com/france24_en.wsx
   or:
   rtsp://server:port/channel_name
4. Open the "More..." tab, and make sure that the Mime type is set to "video/x-ms-asf";
5. Close the dialog and apply/save the changes.

Notes:

- the path portion of the RTSP server depends on the server configuration;
- the default port for the RTSP protocol is 5554;
- the selected server must use one of the video codecs/formats supported by the HMP100, for the streaming to work as expected;
- when username and password are required by the streaming server, the most common format is the following:

protocol://username:password@hostname:port/path

Valid codecs for streaming video

The supported video codecs are:

- MPEG4
- H.264
- WMV 9 (Simple and Main profiles)

The supported audio codec are:

- MP3 @ max 128kbit/s

The recommended codec is MPEG4. Due to its high compression ratio, a video streaming of as low as 500Kbit/s can produce an average quality video stream, while 1Mbit/s produces near TV quality.

For best results the following parameters should be used, if possible:

- video codec: MPEG4, 3Mbit/s,
- audio codec: mp3, 128kbit/s (VBR), 2 channels stereo.

H.264 and Windows Media Video 9 (Simple and Main profiles) are anyway also supported and commonly used.

ADDITIONAL CONFIGURATION

Streaming video buffering

For each video streaming included in a HMD project, the HMP100 allocates a buffer to reproduce the video without interruptions in case of small network hiccups. This value is customizable by adding the following property to the animation tag inside the SVG file:

```
spx:buffering="number_of_seconds"
```

where:

- " number_of_seconds " is the buffer time allocated, in seconds;

The default value for the video buffer is 1.5 seconds.

HARDWARE DEVICES CONFIGURATION

IP Camera (Axis 207W)

This example shows a typical configuration for an Axis 207W camera to produce a video stream usable in a HMD project. Similar options may be used for cameras with the same features. The pictures below show the minimum settings required to be verified for the camera to work with an HMP100.

Network configuration:

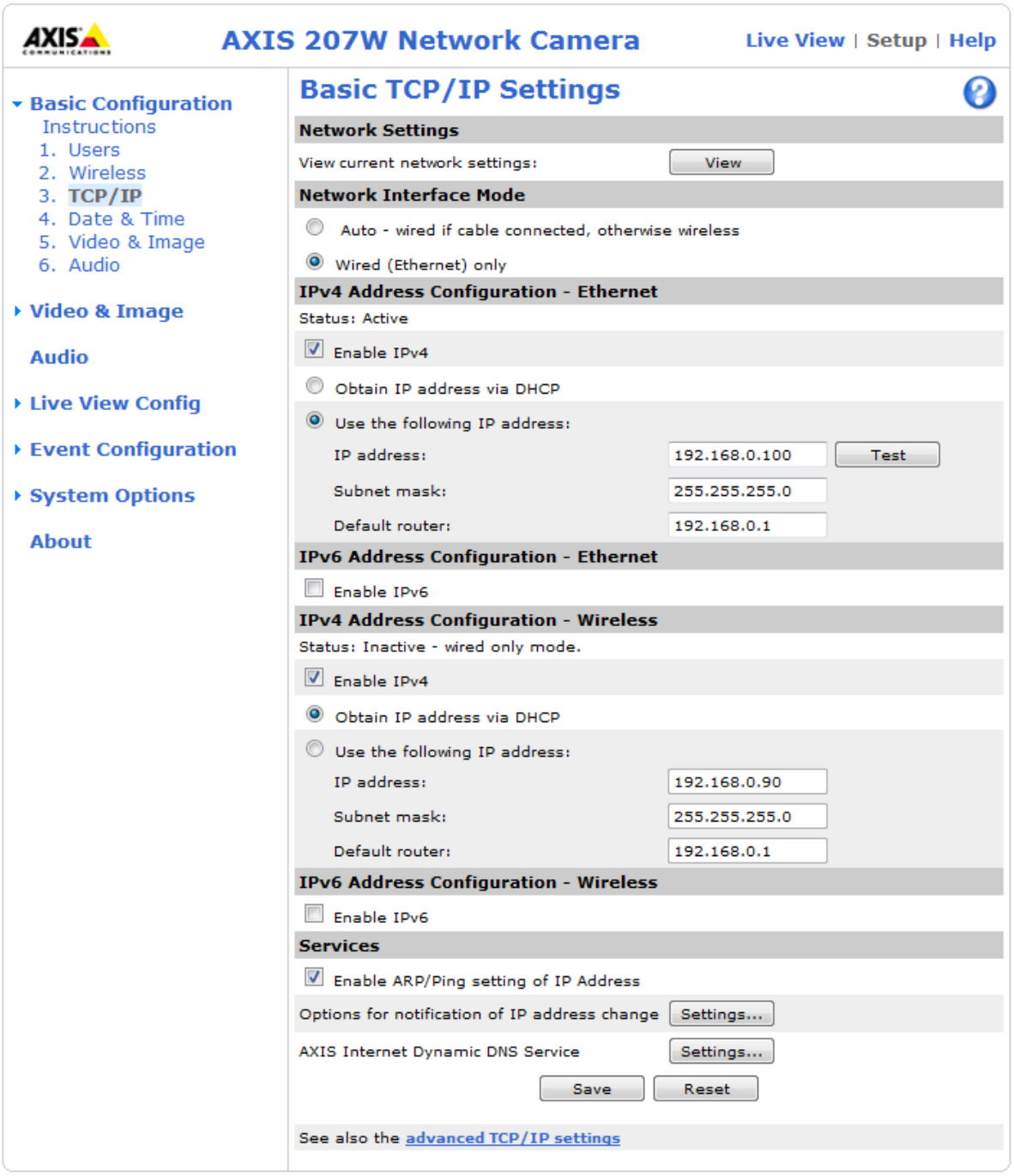
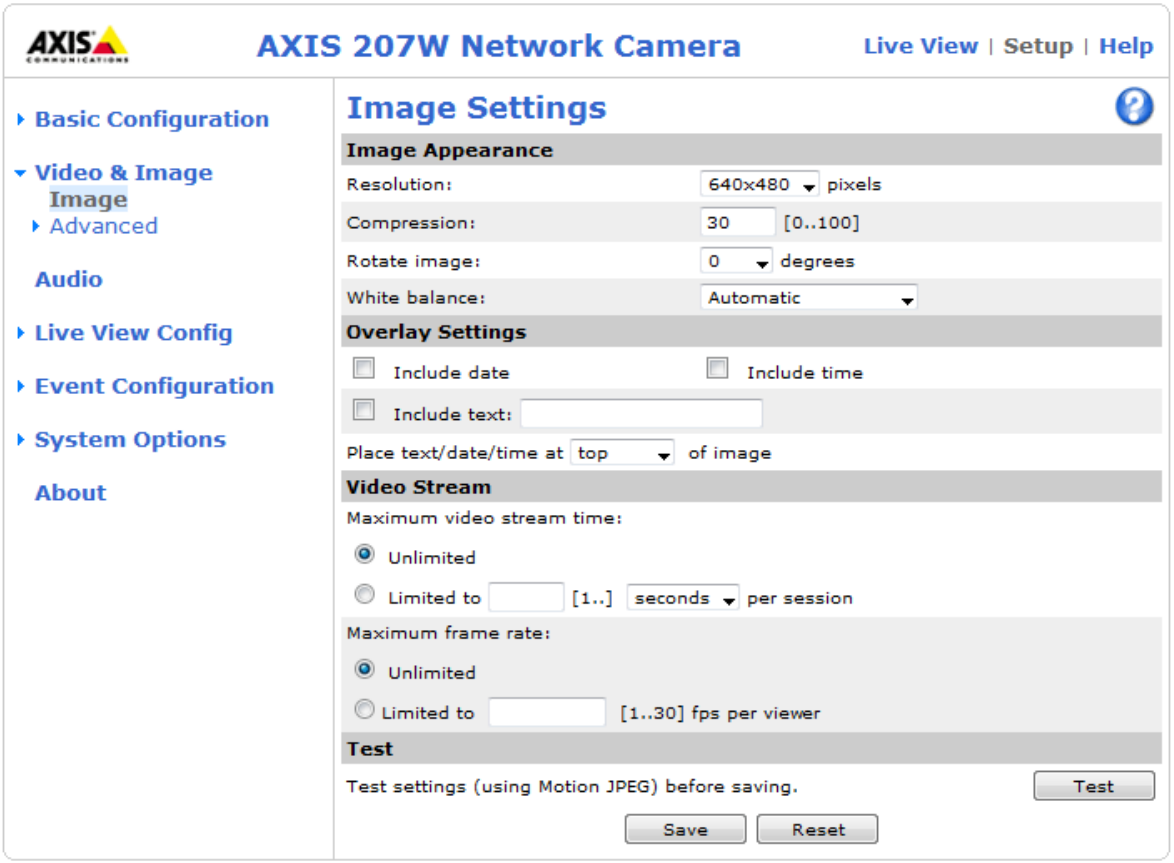
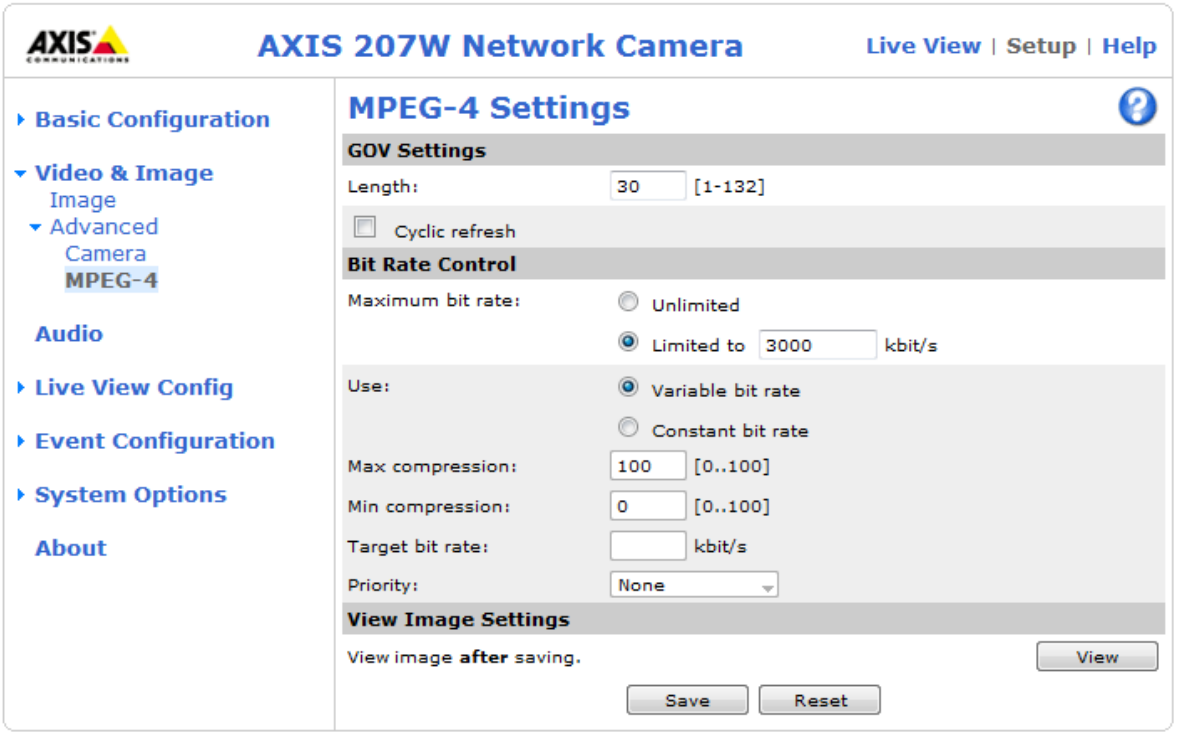


Image configuration:



MPEG4 configuration:



The following is an example of ASX file needed to include the video stream coming from the camera specified earlier into the HMD project:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generator: SpinetiX Hyper Media Director -->
<ASX VERSION="3.0">
 <ENTRY>
  <REF href="rtsp://192.168.0.100:554/mpeg4/media.amp"/>
 </ENTRY>
</ASX>
```

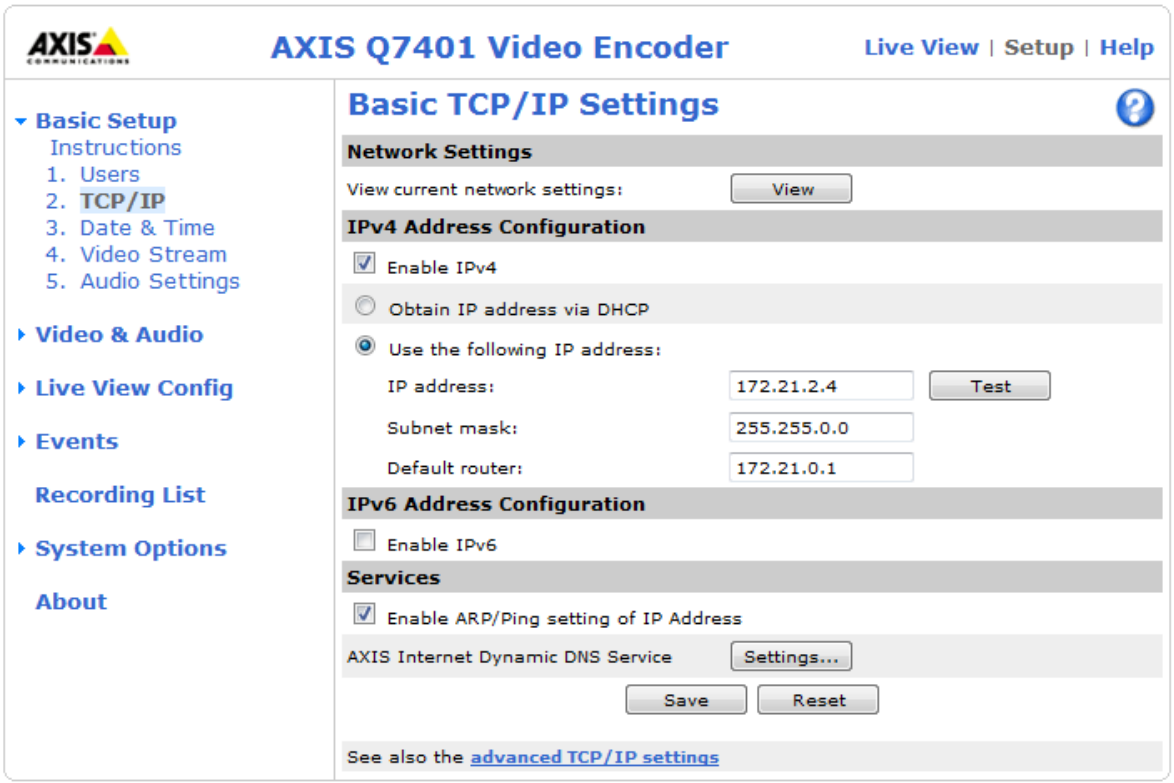
where:

- "rtsp://" is the protocol used by the video streamer;
- "192.168.0.100:554" is the IP address of the video streamer and its port;
- "/mpeg4/media.amp" is the path to request the specific video stream;

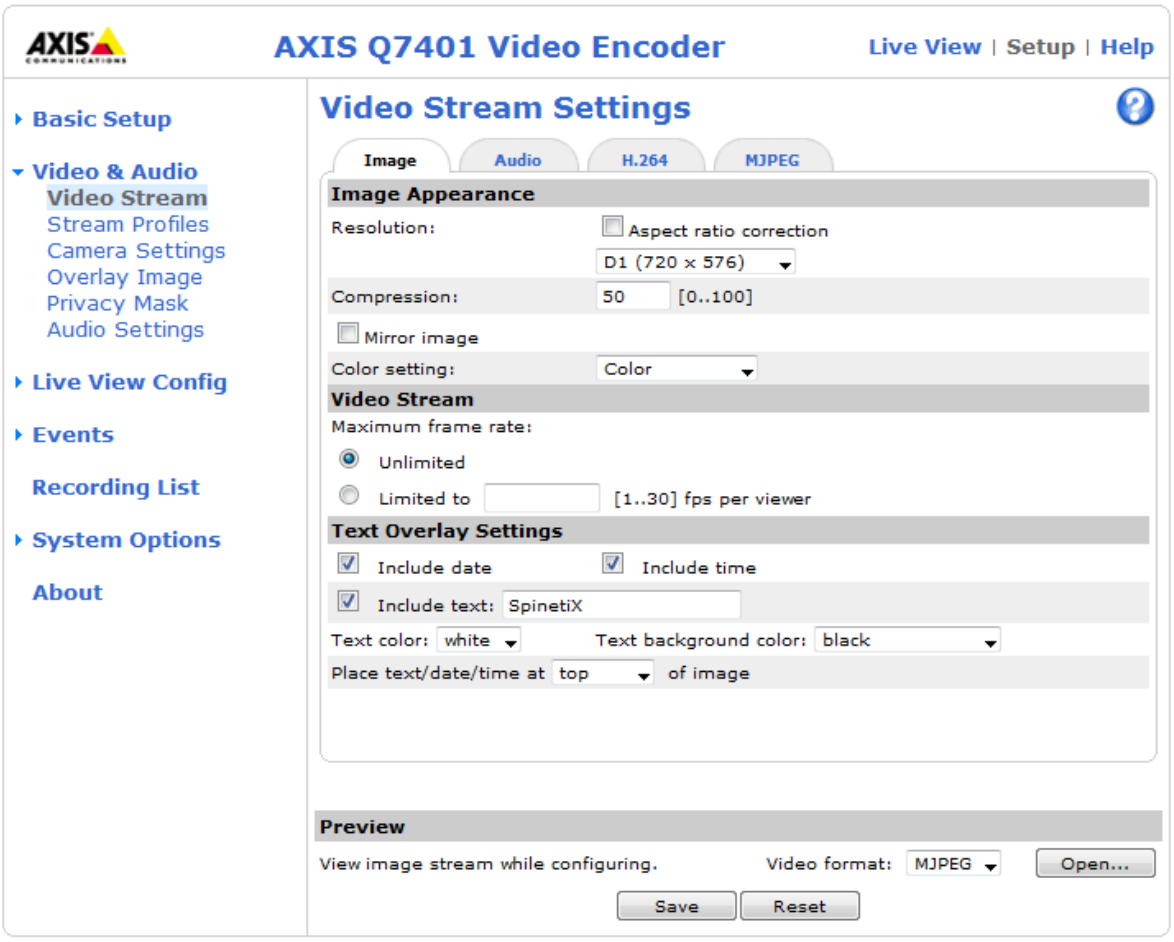
Video Encoder (Axis Q7401)

This example shows a typical configuration for an Axis Q7401 video encoder to produce a video stream usable in a HMD project. Similar options may be used for video encoders with the same features. The pictures below show the minimum settings required to be verified for the encoder to work with an HMP100.

Network configuration:



Video stream configuration:



The following is an example of ASX file needed to include the video stream coming from the encoder specified earlier into the HMD project:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generator: SpinetiX Hyper Media Director -->
<ASX VERSION="3.0">
 <ENTRY>
  <REF href="rtsp://172.21.2.4/axis-media/media.amp"/>
 </ENTRY>
</ASX>
```

where:

- "rtsp://" is the protocol used by the video streamer;
- "172.21.2.4" is the IP address of the video streamer;
- "/axis-media/media.amp" is the path to request the specific video stream;

VIDEOLAN VLC

The "VideoLAN VLC" application

This is an open source software application available at:

http://www.videolan.org/vlc

It is capable of receiving and playing streams but also of producing video streams to be sent to the network. The source of the streams can be any local or remote file, or a video capture device installed on the same PC (TV cards, video capture cards, cameras, etc.)

This section shows how to use VLC as a PC-based, software streamer.

Command line parameters

The easiest way of configuring VLC with the desired parameters is launching it from the command line. One or more Windows batch files (.bat) could also be used to accelerate the launch of VLC with several preset parameters.

The first example makes VLC stream the selected local video over HTTP in a loop:

```
path_to_vlc_folder/vlc.exe selected_video.xyz -L --sout
"#std{access=http,mux=ps,dst=local_IP:local_port/file}
```

where:

- path_to_vlc_folder is the path on your PC to the VLC executable;
- selected_video.xyz is the name of the video file to be streamed;
- local_IP is the local IP address of the PC where VLC is running;
- local_port is the local port number where VLC will stream.

The second example streams the selected file after having trans-coded it into optimal format for the HMP100: MPEG4, 3Mbit/s video codec and mp3, 128kbit/s, stereo audio.

```
path_to_vlc_folder/vlc.exe selected_video.xyz -L --sout
"#transcode{vcodec=DIV3,vb=3072,scale=1,acodec=mp3,ab=128,channels=2}:std
{access=http,mux=asf,dst=local_IP:local_port/file}"
```

where:

- path_to_vlc_folder is the path on your PC to the VLC executable;
- selected_video.xyz is the name of the video file to be streamed;
- local_IP is the local IP address of the PC where VLC is running;
- local_port is the local port number where VLC will stream.

ASX file parameters

The corresponding .asx file to be included into a HMD project, for the examples above would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generator: SpinetiX Hyper Media Director -->
<ASX VERSION="3.0">
 <ENTRY>
  <REF href="http://local_IP:local_port/file"/>
 </ENTRY>
</ASX>
```

where:

- local_IP is the IP address of the PC where VLC is running (same as in the above examples);
- local_port is the local port number where VLC is streaming (same as in the above examples).

TROUBLESHOUTING

If you can't see the video of your streamer on the HMP100, or if you are unsure whether the stream is working at all or not, you may start troubleshooting using a client application on your PC.

Testing stream on a PC

A good and open source choice is VideoLAN VLC (available for download at http://www.videolan.org/vlc).

Try opening the stream from VLC ("Media/Open Network..." option), using the protocol, IP address, port, path/filename provided by your streamer. Pay attention to the difference of URL to use in VLC and on the HMP100:

VLC:	rtp://@	or	udp://@
HMD/HMP100:	rtp://	or	udp://

E.g.:

VLC:	rtp://@224.0.0.1:1234
HMD/HMP100:	rtp://224.0.0.1:1234

You should be able to see the video playing on your PC. If this isn't the case there are probably some issues with the streamer, verify IP address, port, protocol, and then firewall and router configurations.

When you manage to receive and see the video, then you can take the exact URL string used by VLC and paste it into the .asx file mentioned in the previous chapter. Start by trying the .asx file in HMD and then publish your project on the HMP100 for final testing.

Common issues

Typical difficulties in displaying a stream on the HMP100 are due to:

- Firewalls: a firewall could block the video stream coming from the streamer; it could be installed on the streaming source (in case of a PC), on the network premises (router or hardware firewall) or on the PC where HMD is installed and where the testing is performed.

- Router configurations: depending on the network architecture and router(s) configuration, some streams could be blocked; e.g. multicast streams, UDP flows, specific ports, etc. Also, NAT networks are often the cause of the impossibility to have a direct IP connection between the streamer and the player.

All these aspects should be taken into consideration when designing the architecture that will enable the video streaming feature.

Technical Support

In case of unresolved issues please refer to the SpinetiX technical support (support@spinetix.com), providing as many details as possible on: the streamer model and configuration, the network architecture, the URL or ASX file used. In addition enclose

the HMP100 report that can be downloaded from the web administration interface, going to the Maintenance page and clicking on the "Get report" button.