



Tech Note

Working the JPEG way with BVIP



- ▶ Easy integration to 3rd party applications
- ▶ Simple Web browser control
- ▶ OS platform independent
- ▶ Encapsulated CGI commands

Bosch IP Video products can stream MPEG-4 video as well as JPEG images. The JPEG images might be particularly useful if the video management system cannot decode the MPEG-4 streams, or the units need to feed another systems, like Access Control or Alarm Verification systems, where only low frame rates may be sufficient.

The unit's Web browser interface, used to configure the units, view live or playback video, also has a way to view JPEG images.

This tech note offers some idea of how JPEG images can be obtained from the units via HTTP and the newly introduced Common Gateway Interface (CGI). Common uses include a Web browser, a Web server-based application, or in a third party software application.

The examples in this note assume the following devices and their IP addresses:

- a Dinion IP camera 160.10.0.1,
- an Autodome IP camera 160.10.0.4,
- a VideoJet X40 encoder 160.10.0.40 with 4 analog cameras connected.

Taking a JPEG snapshot

JPEG from an IP camera

To get a snapshot image from the Dinion IP use the following URL:

`HTTP://160.10.0.1/snap.jpg`

This will show a snapshot of the camera at the default CIF image size and a default quality setting.



(CIF, 23 kB)

Similarly, if the unit is password-protected, user name and password must be provided, for example:

`HTTP://user:myuserpwd@160.10.0.1/snap.jpg`

For simplicity the following examples assume no protection.

The units can deliver different image sizes by providing the parameter “JpegSize” with one of the following possible values:

S	(small)	176 × 144/120 pixels (QCIF)
M	(medium)	352 × 288/240 pixels (CIF)
L	(large)	704 × 288/240 pixels (2CIF)
XL	(extra large)	704 × 576/480 pixels (4CIF)

The parameter is added to the URL in a query string appended after a question mark. For example to get the largest image:

HTTP://160.10.0.1/snap.jpg?JpegSize=XL

Please note that all parameter names in the query string are case-sensitive.

JPEGs from multi-channel units

A specific camera in a multi-channel unit can also be selected using the “JpegCam” parameter. For example, to connect to video from the second camera of the VideoJet X40:

HTTP://160.10.0.40/snap.jpg?JpegCam=2

Multiple parameters can be handed over equally to standard query string notation in an URL, separated by “&”. The order of the parameters in the query string is not relevant.

Combining the highest resolution image from the second camera on our VideoJet X40, using 4CIF resolution, the URL is:

HTTP://160.10.0.40/snap.jpg?JpegCam=2&JpegSize=XL

Tweaking the JPEG quality

The units use a pre-selected quality setting for all JPEG requests, which defaults to 4 and resembles a fairly good quality.

The required parameter for tweaking this is called “JpegQuality” and has a range from 1 (best) to 32 (modest).

In some applications, e.g. when requiring a small-sized QCIF image only, it could be possible to decrease the quality without losing details but save bandwidth and thus transmission time.

In such a case, taking a low-quality QCIF snapshot from the Dinion IP, the URL could be:

HTTP://160.10.0.1/snap.jpg?JpegSize=S&JpegQuality=32



(QCIF, 9 kB)

Alternatively, for a large-scaled 4CIF display, you might want to increase the quality to make use of the maximum details available. In this case, taking a highest-quality 4CIF snapshot from the Autodome IP, the URL would be:

HTTP://160.10.0.4/snap.jpg?JpegSize=XL&JpegQuality=1



(4CIF, 119 kB)

JPEGs in multi-view

It is possible to get a JPEG containing up to 4 cameras, as long as they are part of a multi-channel unit.

Each of the video sources is identified by a single bit. Camera 1 is represented by bit 0, camera 2 by bit 1, camera 3 by bit 2 and camera 4 by bit 3. These bits are combined into a decimal value as the parameter “JpegCamBits”.

For example, a quad-view of all 4 cameras would use all four bits set to 1, creating a value of 1111 in binary, 0xF in hexadecimal, or 15 in decimal.

Multi-view JPEG images are always scaled to 4CIF width (704 pixels), so the parameter JpegSize is not required. A single camera displayed in one row is scaled to 2CIF.

For example, a quad-view JPEG image in 4CIF resolution from our VideoJet X40 would result in the following URL:

HTTP://160.10.0.40/snap.jpg?JpegCamBits=15

This will produce an image like this:



To improve the appearance you can insert a border around the camera images. The width and color of this border can be defined but note that the borders add to the image size.

The color is defined as hexadecimal values for Y'UV. Y represents the luminance value (the black-and-white component) and basically defines the brightness of the border color, where 0x00 is dark and 0xff is bright. U and V represent the so-called chrominance (color) values where U defines the blue-luminance difference value, and V the red-luminance difference value.

For those who want to gain deeper knowledge about Y'UV and the color space definitions there is a pretty good explanation in Wikipedia.

These 3 bytes are followed by a two-digit value for the border width in steps of 16 pixels, resulting in values of 0x10, 0x20, ... 0xf0. The border is drawn at the top and the right side of each of the single camera images. This allows to seamlessly fit multiple JPEGs from various units on one larger screen.

To set the border color properly you either need some understanding of the Y'UV colour scheme – or you can simply play around with it a bit.

For example, if you want to draw a large red border you should set the Y value to a modest brightness, let's say 0x40, put in a balanced value for blue against green, let's say also 0x40, put the V value to a red-saturated 0xff, and set the border width to 0x80.

The resulting URL should look like:

```
HTTP://160.10.0.40/snap.jpg?JpegCamBits=15&JpegBorder=0x4040ff80
```

This will produce the unattractive image below:



So we will define our border color as a medium-grey as this is least annoying. The values shall be modest 0x80 for all Y, U and V. The border shall be 16 pixels.

Based on this example, a quad-view JPEG image with the defined border settings would result in the following URL:

```
HTTP://160.10.0.40/snap.jpg?JpegCamBits=15&JpegBorder=0x80808010
```

This will produce an image like this:



“Streaming” JPEG

JPEGs continuously updated

Video management systems typically provide live viewing of cameras. In many cases a frame rate lower than 25/30 images per second is quite sufficient. These cases could be covered with a “quick” update of the JPEG image.

If this update is a kind of automated or scheduled, it effectively becomes fast moving pictures, and we call this “Motion JPEG”.

There is no public standard yet that defines the transmission of a “Motion JPEG” stream, so you find many different implementations to achieve a “quasi”-video impression.

BVIP units running firmware 3.0 or later can be used in a mode where almost all the computing performance can be used solely for JPEG creation. As continuous JPEG encoding requires much more performance than MPEG-4 encoding with comparable quality no MPEG-4 encoding would then be possible and also no video content analysis.

Doing so the units can provide up to 25/30 JPEG images per second, which is – from the Motion JPEG view – equal to the maximum frame rate of MPEG-4 video.

But please note that much more data needs to be transmitted than with video as JPEG images can not benefit from the more advanced MPEG-4 compression algorithms.

A management system can implement continuous requesting of updated JPEG images from BVIP units. Some measures must be taken to avoid too many requests that could possibly flood the network if not taken care of.

So there is a mechanism available to avoid transmission of identical JPEGs that already have been sent, e.g. due to too fast repetition of requests.

A viewing system – we will call it “client” – can add an ID to the URL. This ID is defined by the client itself and should be unique throughout the total system, meaning every client shall have its own different ID.

The ID is handed over with the case-sensitive parameter “JpegDomain” and can be any number or alphanumeric string.

