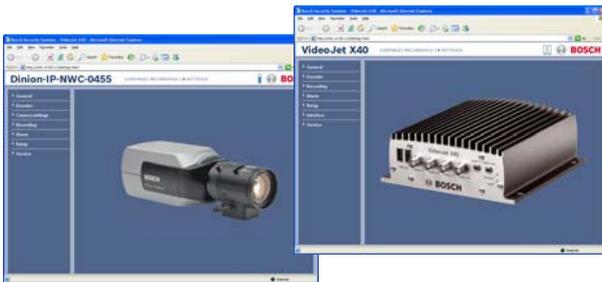




# Tech Note

## Controlling BVIP using CGI commands over HTTP



- ▶ Easy integration into 3<sup>rd</sup> party software application
- ▶ Simple Web browser control
- ▶ 2-way communication
- ▶ OS platform independent
- ▶ All RCP+ commands can be encapsulated in CGI commands

Bosch IP Video products provide a Web browser interface that can be used to configure the units, view live or playback video via the Web browser but naturally can also be controlled and managed by remote management systems.

This tech note shall give a few hints of what could be done with the units via HTTP and the newly introduced Common Gateway Interface (CGI), e.g. with a Web browser, a Web server-based application, or in a “foreign” environment, meaning third party software.

The following examples assume having installed

- a Dinion IP camera at IP address 160.10.0.1,
- an Autodome IP camera at IP address 160.10.0.4,
- a VideoJet X40 at IP address 160.10.0.40 with 4 cameras connected,
- and a VIP XD decoder at IP address 160.10.0.100.

### Direct mapped CGI commands

For simple functions the units provide direct-mapped CGI commands, normally used with the Web browser. Of course, these functions can be password-protected so that the unit can't be accessed or disturbed by anyone unauthorized.

### Resetting a BVIP device

To reset a unit, e.g. the Dinion IP, the URL must look like:  
*HTTP://160.10.0.1/reset*

In case the unit is password-protected, user name and password must be provided as suffixes to the IP address, making the URL look like:  
*HTTP://service:mypasswd@160.10.0.1/reset*

### Taking a JPEG snapshot from a BVIP device

To get a snapshot image from the Dinion IP the URL must look like:

*HTTP://160.10.0.1/snap.jpg*

This will show a snapshot of the camera in the default image size which is CIF.

Similarly, if the unit is password-protected, user name and password must be provided, making e.g. the URL look like:

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

This is further not assumed in our examples.

Sometimes it is preferred or necessary to have a different image size. For that, the units can also provide different formats providing the case-sensitive parameter “JpegSize” with one of the possible values.

Here is the list:

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

The parameter is added to the URL in a query string appended after a question mark.

For receiving a JPEG image in maximum resolution the URL must look like:

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

The video inputs and the respective cameras of multi-channel units can also be selected by handing over a parameter with the snap.jpg command to the VideoJet X40, this time the also case-sensitive parameter “JpegCam”.

To connect to video from the second camera of the VideoJet X40, the URL must look like:

*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.

Expecting that we also want to see the best resolution image from the second camera on our VideoJet X40, using 4CIF resolution, the URL must look like:

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

## RCP+ via CGI commands

Bosch IP Video products in video management system environments are mainly controlled and managed using an enhanced version of Remote Control Protocol, RCP+, pronounced “RCP plus”.

This protocol defines commands and messages that allow to configure the units and to establish communication between units, or units and management systems.

As RCP+ message and command sets are dependent on firmware features the documentation is updated on every firmware release.

Since firmware version 3.0, RCP+ commands can be encapsulated into CGI requests. The command structure of RCP+ must be translated into parameters and values handed over to the unit’s XML-based CGI interpreter as a query string.

Using HTTP as transportation protocol, e.g. with a Web browser, a URL may look like:

```
HTTP://160.10.0.1/rcp.xml?<<query_string>>
```

Each RCP+ command is a fixed predefined 16 bit value followed by a structured set of parameters, some mandatory, some optional. The RCP+ command and the required structure can be taken from the RCP+ documentation. All message elements in the query string must resemble an appropriate pair of parameter and value.

The following examples shall give a principle overview of how to use RCP+ commands with CGI.

### Reading parameters

Various parameters can be read from a Bosch IP Video product, e.g. the hardware or firmware version, the name of the camera(s), or the CPU performance values.

To check our Dinion IP’s encoder load, part of the values displayed with the performance bar in the Web browser, the URL must look like:

```
HTTP://160.10.0.1/rcp.xml?command=0x0a07&type=T_DWORD&direction=READ&num=1
```

The return value provides a recent snapshot value of the CPU performance required for encoding video.

The Web browser is able to display the returned XML message in a readable format. The return value is presented in the <result> section and in this example shows a value of 51%.

The XML return message is displayed in the Web browser like this:

```
- <rcp>
- <command>
  <hex>0x0a07</hex>
  <dec>2567</dec>
</command>
<type>T_DWORD</type>
<direction>READ</direction>
<num>1</num>
<idstring />
<payload />
<cltid>0xd2f9</cltid>
<auth>2</auth>
- <result>
  <hex>0x00000033</hex>
  <dec>51</dec>
</result>
</rcp>
```

The return message displayed in the Web browser also contains unused and optional parameters that are represented by an “empty” XML tag in the form of “<xmltag />”. These may be disregarded.

To check our Dinion IP’s camera name the URL must look like:

```
HTTP://160.10.0.1/rcp.xml?command=0x0019&type=P_UNICODE&direction=READ&num=1
```

The XML return message is displayed in the Web browser like this:

```
- <rcp>
- <command>
  <hex>0x0019</hex>
  <dec>25</dec>
</command>
<type>P_UNICODE</type>
<direction>READ</direction>
<num>1</num>
<idstring />
<payload />
<cltid>0xd82d</cltid>
<auth>2</auth>
- <result>
  <str>00 43 00 61 00 6d 00 65 00 72 00 61 00 20 00 31 00 00</str>
</result>
</rcp>
```

The camera name as return value is coded in Unicode which is not clearly visible to humans but easily interpreted by a computer. The Unicode string in this example resembles the string “Camera 1”.

### Setting serial communication speed

Bosch IP Video products offer various communication speed selections to be configured via Web browser or Configuration Manager.

It had happened that external equipment required 38400 bps which is not in the list of possible selections for the serial communication speed in firmware 3.0. Nevertheless, the hardware is capable of supporting it, and the CGI interface can be used to switch the unit to this bit rate.

To switch the VideoJet X40’s serial interface to 38.4 kbps the URL must look like:

```
HTTP://160.10.0.40/rcp.xml?command=0x027e&type=T_DWORD&direction=WRITE&payload=0x9600&num=1
```

The payload parameter gets the bit rate value where the hexadecimal value 0x9600 equals the decimal value 38400.

Either one can be used, so the following URL is equally valid:  
[HTTP://160.10.0.40/rcp.xml?command=0x027e&type=T\\_DWORD RD&direction=WRITE&payload=38400&num=1](http://160.10.0.40/rcp.xml?command=0x027e&type=T_DWORD RD&direction=WRITE&payload=38400&num=1)

The unit returns an XML page visible in the Web browser indicating the execution having been successfully or not.

```
--<rcp>
- <command>
  <hex>0x027e</hex>
  <dec>638</dec>
</command>
<type>T_DWORD</type>
<direction>WRITE</direction>
<num>1</num>
<idstring />
<payload>0x9600</payload>
<cltid>0x8876</cltid>
<auth>2</auth>
- <result>
  <hex>0x00009600</hex>
  <dec>38400</dec>
</result>
</rcp>
```

The selected value can also be checked with a similar but read-only command:

[HTTP://160.10.0.40/rcp.xml?command=0x027e&type=T\\_DWORD RD&direction=READ&num=1](http://160.10.0.40/rcp.xml?command=0x027e&type=T_DWORD RD&direction=READ&num=1)

The unit will return a similar XML message indicating the selected bit rate.

```
- <rcp>
- <command>
  <hex>0x027e</hex>
  <dec>638</dec>
</command>
<type>T_DWORD</type>
<direction>READ</direction>
<num>1</num>
<idstring />
<payload />
<cltid>0x90a7</cltid>
<auth>2</auth>
- <result>
  <hex>0x00009600</hex>
  <dec>38400</dec>
</result>
</rcp>
```

## Establishing a video connection

Video connections from Bosch IP Video encoders can be established by sending commands like CONNECT\_TO (0xffcc) with the appropriate identifiers and selectors in the payload section of the RCP+ command.

In our example we initiate a video connection from the third decoder of our VIP XD to the VideoJet X40's first video input, using the GET method sent to the VideoJet X40:

[http://160.10.0.100/rcp.xml?command=0xffcc&type=P\\_OCTET &direction=WRITE&payload=0xa00a0028000000030301000000000001](http://160.10.0.100/rcp.xml?command=0xffcc&type=P_OCTET &direction=WRITE&payload=0xa00a0028000000030301000000000001)

In the payload we need to identify the destination IP address, the type of stream we want to connect to, and the local and remote coders and lines.

Described in short form, the payload is combined of:

IP address 160.10.0.40	0xa00a0028
reserved 2 bytes	0x0000
MPEG-4 video, substitute connection	0x0003
third decoder on video output 1(the only one)	0x0301
PUT channels unused	0x0000
first available encoder on first video input line	0x0000
GET video channel	0x0001

When done, we should receive a message in XML format which looks like this:

```
- <rcp>
- <command>
  <hex>0xffcc</hex>
  <dec>65484</dec>
</command>
<type>P_OCTET</type>
<direction>WRITE</direction>
<num>0</num>
<idstring />
<payload>0xa00a0028000000030301000000000001</payload>
<cltid>0x08a9</cltid>
<auth>2</auth>
- <result>
  <str>a0 0a 00 28 00 00 00 03 03 01 00 00 00 00 00 01</str>
</result>
</rcp>
```

Be aware that the VIP XD should be in quad-view mode, or the correct decoder should be selected to verify the result visually.

The remote encoder and line settings, both indicating the first available one, would also work for our Dinion IP and the Autodome IP. The other encoders and lines from our VideoJet X40 could be selected in the respective section, e.g.:

second encoder on fourth video input line 0x0204

If the connection fails, the flags indicating the connected media channels are set to 0 in the result section. A response indicating an unsuccessful connection establishment might look like this:

```
- <rcp>
- <command>
  <hex>0xffcc</hex>
  <dec>65484</dec>
</command>
<type>P_OCTET</type>
<direction>WRITE</direction>
<num>0</num>
<idstring />
<payload>0xa00a0028000000030301000000000001</payload>
<cltid>0x03c9</cltid>
<auth>2</auth>
- <result>
  <str>a0 0a 00 28 00 00 00 03 03 01 00 00 00 00 00 00</str>
</result>
</rcp>
```

The video flag is negatively acknowledged in the GET field, the last 2 bytes in the payload section, indicating that no video connection could be established.

This was the easy part of connection handling. But we do not only want to set up connection after connection, substituting the older one though this would be a possible way, we also want to explicitly disconnect established connections. This requires knowledge of the device-internal reference to the connection. So, let's get it.

## Disconnect a connection

All connections from a BVIP device have a Session-ID assigned when established. The Session-ID must be known as a reference to disconnect the stream.

This can be achieved in two ways:

### 1. Register for reply

Before sending the CONNECT\_TO command, we send the CONNECT\_TO message request, which will collect all incoming connects and display them with the Session-ID:

```
http://160.10.0.100/rcp.xml?message=0xffcc&collectms=5000
```

After issuing this request, the reply returns immediately after a connection using the CONNECT\_TO command has been established, or the specified timeout of 5 seconds has expired. Then, within these 5 seconds we use a second Web browser window to send the CONNECT\_TO command as before:

```
http://160.10.0.100/rcp.xml?command=0xffcc&type=P_OCTET
&direction=WRITE&payload=0xa00a00280000000030100001010001
0000001
```

When done, we should receive a message - in the Web browser window where we are waiting for it - in XML format which looks like this:

```
- <message_list>
- <msg>
  <command>0xffcc</command>
  <num>0</num>
  <sessionID>0xca2e0862</sessionID>
  <hex>0xa00a0028000000000030100001010001</hex>
</msg>
</message_list>
```

Thus, we got the Session-ID (0xca2e0862) and can disconnect the stream with:

```
http://160.10.0.100/rcp.xml?command=0xff0d&type=P_OCTET
&direction=WRITE&sessionid=0xca2e0862
```

```
- <rcp>
- <command>
  <hex>0xff0d</hex>
  <dec>65293</dec>
</command>
<type>P_OCTET</type>
<direction>WRITE</direction>
<num>0</num>
<idstring />
<payload />
<cltid>0x172e</cltid>
<auth>2</auth>
- <result>
  <str>01 00 00 00 a0 0a 00 28</str>
</result>
</rcp>
```

The first byte 0x01 indicates that the connection was successfully disconnected.

### 2. Check connection list

The other way to get the Session-ID is to display all active connections, and we have to pick the right one for disconnecting.

To achieve this, we send the command CONF\_ACTIVE\_CONNECTION\_LIST (0xffc1) via CGI:

```
http://160.10.0.100/rcp.xml?command=0xffc1&type=P_OCTET
&direction=READ
```

Assuming we had already established 3 connections, we should get a reply similar to this:

```
- <rcp>
- <command>
  <hex>0xffc1</hex>
  <dec>65473</dec>
</command>
<type>P_OCTET</type>
<direction>READ</direction>
<num>0</num>
<idstring />
<payload />
<cltid>0x1cfe</cltid>
<auth>2</auth>
- <result>
  <str>
a0 0a 00 04 01 01 00 04 ca 28 00 33 01 01 00 00 00 00 00 00 00 00 01
a0 0a 00 01 02 01 00 04 ca 2e 08 63 01 01 00 00 00 00 00 00 00 00 01
a0 0a 00 28 03 01 00 04 ca 2e 08 64 01 03 00 00 00 00 00 00 00 00 01
  </str>
</result>
</rcp>
```

The returned result string is a list of the 3 connections which is already separated into 3 lines for better visibility in the example above.

We can see from the leading IP address that we have

- one connection to the Autodome IP on our first decoder,
- one to the Dinion IP on the second decoder,
- and last the interesting one from the VideoJet X40 to the third decoder, correlated to Session-ID 0xca2e0864.

Again, we got our Session-ID and can disconnect the stream with:

```
http://160.10.0.100/rcp.xml?command=0xff0d&type=P_OCTET
&direction=WRITE&sessionid=0xca2e0864
```

<b>Americas:</b> Bosch Security Systems 130 Perinton Parkway Fairport, New York, 14450, USA Phone:+1 585 223 4060 Fax:+1 800 289 0096 security.sales@us.bosch.com http://www.boschsecurity.us	<b>Europe, Middle East, Africa:</b> Bosch Security Systems B.V. P.O. Box 80002 5600 JB Eindhoven, The Netherlands Phone:+31 (0) 40 27 83955 Fax:+31 (0) 40 27 86668 emea.securitysystems@bosch.com http://www.boschsecurity.com	<b>Asia-Pacific:</b> Bosch Security Systems Pte Ltd 38C Jalan Pemimpin Singapore 577180 Phone:+65 6319 3450 Fax:+65 6319 3499 apr.securitysystems@bosch.com http://www.boschsecurity.com
--	--	---

Data subject to change without notice | 11 April 2008