

ADAPTIVE RECOGNITION

ANPR/ALPR CAMERAS AND SOFTWARE FOR TRAFFIC MONITORING

# Programmer's Guide

# ANPR/ALPR Cameras



# ANPR/ALPR cameras

## Programmer's Guide

Document version: v.3.3.

### Table of Contents

1.	General API Specification.....	4
1.1.	Sections .....	4
1.2.	Parameters .....	4
1.3.	Parameter Attributes .....	6
1.4.	Answer Document.....	10
2.	Modules.....	11
2.1.	Users.....	11
2.2.	Image Resolution .....	12
2.3.	Simple Retrieval of an Image .....	13
2.4.	Image/Image Parameter Queries.....	13
2.5.	Frequently Used Image Parameters .....	15
2.6.	Capture Parameters.....	16
2.7.	Auto Brightness Control.....	16
2.8.	Auto Focus .....	17
2.9.	Auto Color Compensation .....	18
2.10.	Optics Control .....	19
2.11.	Flash Control.....	20
2.12.	Event Manager.....	21
2.13.	Motion Detection Trigger.....	23
2.14.	Software Trigger.....	24
2.15.	UART Trigger.....	25
2.16.	RADAR Trigger (VIDAR WITH RADAR ONLY) .....	26
2.17.	GPIO Trigger.....	26
2.18.	Scheduler Trigger .....	27
2.19.	Plate Finder .....	27
2.20.	Laser trigger .....	27
2.21.	Upload Manager (Non-smart cameras only) .....	28
2.22.	Set/Get ANPR Properties (SMART CAMERAS ONLY).....	30
2.23.	Browse the Database (SMART CAMERAS ONLY) .....	31



2.24.	Database contents (SMART CAMERAS ONLY).....	33
2.25.	Query Database (SMART CAMERAS ONLY) .....	39
2.26.	Templates (SMART CAMERAS ONLY).....	43
2.27.	Engine Manager (SMART CAMERAS ONLY) .....	47
2.28.	Result Upload (only smart cameras).....	57
2.29.	Clear Database/Remove Record (SMART CAMERAS ONLY).....	59
2.30.	ANPR Module Restart (SMART CAMERAS ONLY) .....	59
2.31.	System Information.....	60
2.32.	Camera Log.....	61
2.33.	Black/Whitelist (SMART CAMERAS ONLY).....	62
2.34.	Restart.....	65
2.35.	Camera Modules.....	65
CONTACT INFORMATION.....		66



# 1. GENERAL API SPECIFICATION

Every operation on the VIDAR camera is done by setting and getting parameters. Camera functions are implemented by modules, modules are grouped in module groups. Camera parameters are organized in named sections within modules. Parameters are set and get by HTTP queries in the manner detailed in this chapter

## 1.1. SECTIONS

Querying available parameter sections:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?getsections&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?getsections&wfilter=X)

ANSWER:

```
mimetype=text/plain\r\n
sections=default,sectionname1_sectionvalue1,sectionname2_sectionvalue2,..
.\r\n
```

## 1.2. PARAMETERS

Query of the parameter values:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?get\[parameter1\\_name\]&get\[parameter2\\_name\]&...&\[sectionname\]=\[sectionvalue\]&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?get[parameter1_name]&get[parameter2_name]&...&[sectionname]=[sectionvalue]&wfilter=X)

ANSWER:

```
...
parameter1_name=parameter1_value\r\n
parameter2_name=parameter2_value\r\n
...
```

### Note

- If there is no section, the default section is applied
- One or even more parameters can be queried (if these belongs to one section)

Querying values of all parameters of the default section:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?getall&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?getall&wfilter=X)

ANSWER:

```
mimetype=text/plain\r\nsections=default\r\nparametername1=parametervalue1\r\nparametername2=parametervalue2\r\n...
```

Querying values of all parameters of the given section:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?getall&\[sectionname\]=\[sectionvalue\]&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?getall&[sectionname]=[sectionvalue]&wfilter=X)

ANSWER:

```
mimetype=text/plain\r\nparametername1=parametervalue1\r\nparametername2=parametervalue2\r\n...
```

#### Note

- Only one section can be queried at a time.

Setting parameter values:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?\[parametername1\]=\[parametervalue1\]&\[parametername1\]=\[parametervalue1\]&...&\[sectionname\]=\[sectionvalue\]&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?[parametername1]=[parametervalue1]&[parametername1]=[parametervalue1]&...&[sectionname]=[sectionvalue]&wfilter=X)

ANSWER:

```
mimetype=text/plain\r\necode=0\r\n
```

ANSWER:

```
mimetype=text/plain\r\necode=-22\r\neparams=parametername1,parametername1\r\n
```

**Note**

- Add &save to the end of the query to have the parameters saved and thus made permanent, otherwise the changes will be lost upon restarting the camera
- If section is not defined, the default section will be modified
- More parameters (in the same section) can be get/set
- In the answer the following fields will be present:
  - If no error occurred:  
ecode = 0
  - If error occurred:  
ecode= Standard Error Code  
eparams=parametername1,parametername2,...

### 1.3. PARAMETER ATTRIBUTES

Query of parameter attributes:

GET:

[http://\[camera\\_ip\]/\[modulegroup\]/\[modulename\]?query\[parametername1\]&query\[parametername2\]&...&\[sectionname\]=\[sectionvalue\]&wfilter=X](http://[camera_ip]/[modulegroup]/[modulename]?query[parametername1]&query[parametername2]&...&[sectionname]=[sectionvalue]&wfilter=X)

**Note**

- If there is no section defined, the default section is applied
- More parameters (of the same section) can be queried

[http://camera\\_ip/modulegroup/modulename?queryall\[&sectionname=sectionvalue\]&wfilter=X](http://camera_ip/modulegroup/modulename?queryall[&sectionname=sectionvalue]&wfilter=X)

**Note**

- If there is no section defined, the default section is applied

ANSWER:

- B - bound: B[parametername]=minimum,maximum,default,step
- L - list: L[parametername]=value1,value2,value3,.. (where value1 is the default value)
- T - type

EXAMPLE

GET:

[http://camera\\_ip/capture/scapture?queryshutter&wfilter=X](http://camera_ip/capture/scapture?queryshutter&wfilter=X)

EXAMPLE – ANSWER:

```
...
Bshutter=25,33000,1000,0.001\r\n
Tshutter=rw,f,us\r\n
...
```

The bound answer:

bound[parameter\_name]=minimum,maximum,default,step

- **minimum** value of the specified property (integer or floating-point number)
- **maximum** value of the specified property (integer or floating-point number)
- **default** (or recommended) value
- the amount of the steps to be set / advisable to set (helpful for programming)

#### Note

- Some/all of the fields minimum, maximum, default and step may be missing. This means the corresponding value is undefined.

Example for the bound:

- boundshutter=20,10000,1000,1

The value of the shutter can be set between 20 and 10000 in steps of 1, the recommended value is 1000.

- boundshutter=20,10000, ,1

The value of the shutter can be set between 20 and 10000 in steps of 1, there is no recommended value.

- boundshutter=20,10000,1000

The value of the shutter can be set between 20 and 10000, the recommended value is 1000. There are no steps, so in case of integer it is advisable to step one by one.

- boundshutter=20,10000

The value of the shutter can be set between 20 and 10000. There are no recommended value and steps.

**The list answer:**

L[parameter\_name]=value1,value2,value3,.. (where value1 is the default value)

- Possible values in the list (the recommended value should be the first). The values can be integers or floating-point number and strings.

L[parameter\_name]=\$

- The value of the property is not examined (e.g.: username). Return of property, where neither the bound nor the list can be used.

**Example for the list:**

- Lifps=22.5,11.25

The set FPS value can be 22.5 and 11.25.

The recommended value is 22.5 (the first).

- Lifps=30,10,20,25,45,50,60

The settable FPS values: may be 10,20,25,30,45,50 and 60.

The recommended value is 30 (the first).

- Lcapture=prev,next,best,local

The prev, next, best or local values can be specified for the capture parameter.

- Lpassword=\$

The password parameter can be whatever string (no listing).

**The type answer:**

T[parameter\_name]=access,type,unit,reboot

- access: the mode of the access
  - r: read only
  - w: write only
  - rw: read-write
- type: type of the parameter
  - i8, i16, i32, i64: signed integer 8, 16, 32 and 64 bit type (default value: 32)
  - u8, u16, u32, u64: unsigned integer 8, 16, 32 and 64 bit type (default value: 32, only u is defined)
  - f32, f64: floating-point number (default value: 32, only f is defined)
  - s[number of the characters]: string type with specified character number (if it is not specified, the default will be 32). In case of 0 there are minimum and maximum, which is written in the bound.
  - With all these types a square bracket may be present, which means that an array should be used (e.g.: the i16[4] means 4 x 16-bit integer numbers).
- unit: unit of measurement (optional) - arbitrary string



- reboot: should the system be rebooted after rewriting the parameter
  - not specified: no need to restart
  - q: quick module restart (see Restart)
  - f: complete reboot (see Restart)

#### Example for the type:

- Tshutter=rw,f,us

The shutter is readable and writable, floating-point type and its unit is microseconds. No restart is needed after rewriting the parameter, because it will be entered immediately.

- Timgshutter=r,f,us

The imgshutter is a read only, floating-point type property and its unit is microseconds.

- Tusername=rw,s32,,f

The username is a readable and writable string type containing 32 characters. It does not have measurement, but the system has to be restarted in case of switching.

## 1.4. ANSWER DOCUMENT

A HTTP query must be assembled according to the previous chapters. By default, the answer is a multipart document, consisting of a binary and a text part (e.g. an image in the binary and its parameters, like shutter gain, etc. in the textual part). A special parameter 'wfilter' may be used to control this behaviour:

- wfilter=1 only the text part,
- wfilter=2 only the binary part,
- wfilter=3 both parts will be returned by the camera.

The HTTP header gives information about parsing the message. The 'mimetype' field determines the type of the data in the answer.

- mimetype=application/octet-stream: general data
- mimetype=text/plain: standard text type (e.g. parameters)
- mimetype=image/jpeg: JPEG image

Parameters are always sent in text/plain format, row by row as [name]=[value] pairs. Rows are terminated by '\r\n'. For example:

```
contrast=1\r\n
shutter=1000\r\n
gain=4\r\n
```

Parts in the multipart message are separated by a boundary string, for example:

```
-----7327659421310868402068919661
```

After the separator the length and the type of the part will be written:

```
-----7327659421310868402068919661
```

```
Content-Length: 93123
```

```
Content-Type: image/jpeg
```

The boundary will be written at the beginning of the answer, between the parts and at the end of the answer.

## 2. MODULES

### 2.1. USERS

**Module:** setup/users

**Manual:** Ch. 5.3 Users

Only administrators have access to this module.

Users are described by the following parameters:

- user id (uid)
- user name (uname)
- full name (ufull)
- access level (ulevel)
- password (passw1,passw2)

#### Get/Set access level of users without credentials

Parameter: guestlevel

Possible values: 0: Guest, 1: Viewer, 2: Normal, 3: Power, 4: Administrator

Example:

[http://camera\\_ip/setup/users?wfilter=X&getall](http://camera_ip/setup/users?wfilter=X&getall)

#### Getting the list of users

Example:

[http://camera\\_ip/setup/users?wfilter=X&getall](http://camera_ip/setup/users?wfilter=X&getall)

In the response for 'getall' we get the 'guestlevel' and lists of the user parameters as comma-separated values. Parameter values with the same position/index in the lists belong to the same user. Passwords cannot be obtained.

Response example:

```
guestlevel=0
useridlist=1001,1002
nicknamelist=user00,user01
fullnamelist=,
userlevellist=2,4
```

### Add new user

To add a new user 'uid' must be set to 'new'. 'passw1' and 'passw2' must match.

[http://camera\\_ip/setup/users?wfilter=X&getall&uid=new&uname=user00&ufull=&ulevel=2&passw1=x&passw2=x](http://camera_ip/setup/users?wfilter=X&getall&uid=new&uname=user00&ufull=&ulevel=2&passw1=x&passw2=x)

### Modify the parameters of a user

To modify the parameters of a user the corresponding 'uid' must be given with the modified parameters. 'passw1' and 'passw2' parameters can be omitted if we don't want to change the password.

[http://camera\\_ip/setup/users?wfilter=X&getall&uid=1001&uname=user007&ufull=&ulevel=2&passw1=y&passw2=y](http://camera_ip/setup/users?wfilter=X&getall&uid=1001&uname=user007&ufull=&ulevel=2&passw1=y&passw2=y)

### Delete user

To delete a user the user id ('uid') and the 'delete' parameter must be given.

[http://camera\\_ip/setup/users?wfilter=X&getall&uid=1001&delete](http://camera_ip/setup/users?wfilter=X&getall&uid=1001&delete)

## 2.2. IMAGE RESOLUTION

**Module:** capture/scapture

**Manual:** Ch. 6.1 Imaging Pipeline

### Get maximum and set resolutions

GET:

[http://camera\\_ip/capture/scapture?getall&wfilter=X\[&source=X\]](http://camera_ip/capture/scapture?getall&wfilter=X[&source=X])

ANSWER:

```
...
imgcapxsize=1280\r\n (maximum)
imgcapysize=960\r\n ...
imgxsize=640\r\n (current)
imgysize=480\r\n
...
```

#### Note

- The source # 1 is the default (primary image source) if there is no source parameter.
- In this property list, each value is listed twice: with and without the 'img' prefix. The parameter value with the 'img' prefix represents the value of the parameter at the time the image was captured, while the one without it represents the actual value.
- In case of two optical systems, each sensor can be controlled separately with the following optional parameter: & sensor = 1 or 2. If this parameter is not specified, the parameters for the main sensor are controlled.

## 2.3. SIMPLE RETRIEVAL OF AN IMAGE

**Module:** capture/scapture

**Manual:** Ch. 6.1 Imaging Pipeline

### Capture an image

GET:

[http://camera\\_ip/capture/scapture?wfilter=0\[&source=X\]](http://camera_ip/capture/scapture?wfilter=0[&source=X])

[http://camera\\_ip/capture/scapture?capture=best&framemonotimems=98741855&wfilter=0\[&source=X\]](http://camera_ip/capture/scapture?capture=best&framemonotimems=98741855&wfilter=0[&source=X])

[http://camera\\_ip/capture/scapture?capture=best&frameindex=8390&wfilter=0\[&source=X\]](http://camera_ip/capture/scapture?capture=best&frameindex=8390&wfilter=0[&source=X])

[http://camera\\_ip/capture/scapture?sensor=2](http://camera_ip/capture/scapture?sensor=2)

#### Note

- The source # 1 is the default if there are no source parameters (primary image source).
- See also [#4.Image/Image Parameter Queries](#)
- In case of two optical systems, each sensor can be controlled separately with the following optional parameter: & sensor = 1 or 2. If this parameter is not specified, the parameters for the main sensor are controlled.

## 2.4. IMAGE/IMAGE PARAMETER QUERIES

**Module:** capture/scapture

**Manual:** Ch. 6.1 Imaging Pipeline

### Get all image parameters

GET:

[http://camera\\_ip/capture/scapture?getall&wfilter=X\[&source=X\]](http://camera_ip/capture/scapture?getall&wfilter=X[&source=X])

[http://camera\\_ip/capture/scapture?queryall&wfilter=X&source=X\]](http://camera_ip/capture/scapture?queryall&wfilter=X&source=X])

[http://camera\\_ip/capture/scapture?getall&wfilter=X&capture=next\[&source=X\]](http://camera_ip/capture/scapture?getall&wfilter=X&capture=next[&source=X])

#### Note

- The source # 1 is the default (primary image source) if there are no source parameters.
- In this property list, each value is listed twice: with and without the 'img' prefix. The parameter value with the 'img' prefix represents the value of the parameter at the time the image was captured, while the one without it represents the actual value.
- In case of two optical systems, each sensor can be controlled separately with the following optional parameter: & sensor = 1 or 2. If this parameter is not specified, the parameters for the main sensor are controlled.

The scapture module returns three parameters for every captured frame: `imgframeindex`, `imgframetimems` and `imgframemonotimems`.

**imgframeindex:** After camera startup, captured frames are counted. This parameter provides the number of the frame.

**imgframetimems:** Time elapsed since 1st of January 1970 until the capturing of the frame in milliseconds.

**imgframemonotimems:** Time elapsed since camera startup until the capturing of the frame in milliseconds.

The above parameters can be used to determine the time of capturing when getting frames/parameters from the scapture module.

### Querying the image buffer

#### **capture=last**

The camera returns the last captured image/parameter.

GET:

[http://camera\\_ip/capture/scapture?capture=last&wfilter=X](http://camera_ip/capture/scapture?capture=last&wfilter=X)

#### **capture=next**

It returns the frame/frame's parameter that follows the last returned (queried) one. If this frame is not in the buffer anymore, (too much time has elapsed and it has been rewritten by newer images), it returns the oldest image from the buffer.

The **capture=next** can be used by itself or together with the `framemonotimems`, `imgframetimems` and `frameindex` parameters: In these cases, frames, following the specified number/timestamp are returned (if they are still in the buffer):

GET:

[http://camera\\_ip/capture/scapture?capture=next&framemonotimems=98741855&wfilter=X](http://camera_ip/capture/scapture?capture=next&framemonotimems=98741855&wfilter=X)

[http://camera\\_ip/capture/scapture?capture=next&frameindex=8390&wfilter=X](http://camera_ip/capture/scapture?capture=next&frameindex=8390&wfilter=X)

#### **capture=best**

It returns the frame/frame's parameter that is closest (before or after) to the specified timestamp or frame index. If the buffer does not contain such a frame, the oldest one is returned. If the specified timestamp (`framemonotimems`, `imgframetimems`) or `frameindex` value is too high (refers to a yet non-existing frame) then the newest frame from the buffer is returned.

GET:

[http://camera\\_ip/capture/scapture?capture=best&framemonotimems=106964188&wfilter=X](http://camera_ip/capture/scapture?capture=best&framemonotimems=106964188&wfilter=X)

[http://camera\\_ip/capture/scapture?capture=best&frameindex=193390&wfilter=X](http://camera_ip/capture/scapture?capture=best&frameindex=193390&wfilter=X)

**Note**

- Use 'wfilter' to get either or both the image and its parameters.

SET:

[http://camera\\_ip/capture/scapture?parametername1=parametervalue1&parametername2=parametervalue2&...&wfilter=X\[&source=X\]](http://camera_ip/capture/scapture?parametername1=parametervalue1&parametername2=parametervalue2&...&wfilter=X[&source=X])

SAVE:

[http://camera\\_ip/setup/profiles?save\\_imgparams&wfilter=1\[&c\\_source=X\]](http://camera_ip/setup/profiles?save_imgparams&wfilter=1[&c_source=X])

## 2.5. FREQUENTLY USED IMAGE PARAMETERS

**Module:** capture/scapture

**Manual:** Ch. 6.1 Imaging Pipeline

### Get image parameters

GET:

[http://camera\\_ip/capture/scapture?getall&wfilter=X\[&source=X\]](http://camera_ip/capture/scapture?getall&wfilter=X[&source=X])

ANSWER:

```
...  
imggamma=1.700\r\n  
gamma=1.700\r\n  
...
```

SET:

[http://camera\\_ip/capture/scapture?gamma=1.0&brightness=0.0&contrast=0.0&jpegquality=75&blacklevel=0.0 &whitelevel=1.0&whitebalance=0.0,0.0,0.0&saturation=0.0&wfilter=1\[&source=X\]](http://camera_ip/capture/scapture?gamma=1.0&brightness=0.0&contrast=0.0&jpegquality=75&blacklevel=0.0 &whitelevel=1.0&whitebalance=0.0,0.0,0.0&saturation=0.0&wfilter=1[&source=X])

SAVE:

[http://camera\\_ip/setup/profiles?save\\_imgparams&wfilter=X\[&c\\_source=1\]](http://camera_ip/setup/profiles?save_imgparams&wfilter=X[&c_source=1])

## 2.6. CAPTURE PARAMETERS

**Module:** capture/scapture

**Manual:** Ch. 6.1 Imaging Pipeline

Get and set capture parameters (shutter, gain, etc.)

GET:

[http://camera\\_ip/capture/scapture?getall&source=1&wfilter=X](http://camera_ip/capture/scapture?getall&source=1&wfilter=X)

ANSWER:

```
...
imggain=2.000\r\n
imgshutter=4000.000\r\n
imgllm=3\r\n
...
gain=2.000\r\n
shutter=4000.000\r\n
llm=3\r\n
...
```

SET:

[http://camera\\_ip/capture/scapture?gain=2.5&shutter=2000&wfilter=X](http://camera_ip/capture/scapture?gain=2.5&shutter=2000&wfilter=X)

SAVE:

[http://camera\\_ip/setup/profiles?save\\_imgparams&wfilter=X](http://camera_ip/setup/profiles?save_imgparams&wfilter=X)

## 2.7. AUTO BRIGHTNESS CONTROL

**Module:** control/brightnesscontrol

**Manual:** Ch. 6.3 Brightness Control

Get/Set Brightness Control Properties

GET:

[http://camera\\_ip/control/brightnesscontrol?getall&wfilter=1](http://camera_ip/control/brightnesscontrol?getall&wfilter=1)

[http://camera\\_ip/control/brightnesscontrol?queryall&wfilter=1](http://camera_ip/control/brightnesscontrol?queryall&wfilter=1)

SET:

1. sensor settings:

[http://camera\\_ip/control/brightnesscontrol?enable\\_0=1&save&wfilter=1](http://camera_ip/control/brightnesscontrol?enable_0=1&save&wfilter=1)

[http://camera\\_ip/control/brightnesscontrol?target\\_0=-20&save&wfilter=1](http://camera_ip/control/brightnesscontrol?target_0=-20&save&wfilter=1)

2. sensor settings:

[http://camera\\_ip/control/brightnesscontrol?enable\\_1=0&save&wfilter=1](http://camera_ip/control/brightnesscontrol?enable_1=0&save&wfilter=1)

[http://camera\\_ip/control/brightnesscontrol?target\\_1=10&save&wfilter=1](http://camera_ip/control/brightnesscontrol?target_1=10&save&wfilter=1)



GET/SET Brightness control ROI (Region of Interest)

[http://camera\\_ip/capture/scapture?wfilter=1&sensor=1&source=1&histowinpc=50,50,100,100](http://camera_ip/capture/scapture?wfilter=1&sensor=1&source=1&histowinpc=50,50,100,100)

The first two values are the coordinates of the upper left corner of the square, the second two values are the coordinates of the lower right corner of the square as a percentage (x1, y1, x2, y2). The position of the origin increases in the upper left corner of the image, with the y axis going down.

## 2.8. AUTO FOCUS

**Module:** control/focuscontrol

**Manual:** Ch. 6.5 Optics

Get autofocus state and properties

GET:

[http://camera\\_ip/control/focuscontrol?getall&wfilter=1](http://camera_ip/control/focuscontrol?getall&wfilter=1)

ANSWER:

```
...  
mode=0\r\n  
roi=0,0,1280,960\r\n  
...
```

SET:

[http://camera\\_ip/control/focuscontrol?roi=0,0,100,200&save&wfilter=1](http://camera_ip/control/focuscontrol?roi=0,0,100,200&save&wfilter=1)

[http://camera\\_ip/control/focuscontrol?mode=1&wfilter=1](http://camera_ip/control/focuscontrol?mode=1&wfilter=1)

### Note

- during focusing mode will equal 1, when the focusing is done resets to 0
- roi: operation area of the autofocus; x1, y1, x2, y2, where x1, y1 the upper left corner of the rectangle and x2, y2 the lower right corner of the rectangle

## 2.9. AUTO COLOR COMPENSATION

**Module:** control/colcorr

**Manual:** Ch. 6.2.2 Color

Get color correction state and properties

GET:

[http://camera\\_ip/control/colcorr?getall&wfilter=1\[&sensor=X\]](http://camera_ip/control/colcorr?getall&wfilter=1[&sensor=X])

ANSWER:

...

mode=0\r\n

...

SET: [http://camera\\_ip/control/colcorr?mode=1&save&wfilter=1\[&sensor=X\]](http://camera_ip/control/colcorr?mode=1&save&wfilter=1[&sensor=X])

where

X=1, sensor 1

X=2, sensor 2

## 2.10. OPTICS CONTROL

**Module:** hwlayer/extopt, control/focuscontrol

**Manual:** Ch. 6.5 Optics

Operate motorized optics (zoom, focus, iris, filter exchanger)

GET:

[http://camera\\_ip/hwlayer/extopt?getall&wfilter=1](http://camera_ip/hwlayer/extopt?getall&wfilter=1)

ANSWER:

```
...
focus_pc=56\r\n
zoom_pc=28\r\n
iris_pc=100\r\n
zfcovement=1\r\n
filter=0\r\n
...
```

### Note

In the case of two optical systems, each optic can be controlled separately with the following optional parameter: & ix = 0 or 1. If this parameter is not specified, the parameters of the optics belongs to the main sensor are controlled.

Add zoom/focus presets

GET:

[http://camera\\_ip/control/focuscontrol?getall&wfilter=1](http://camera_ip/control/focuscontrol?getall&wfilter=1)

ANSWER:

```
...
zfc_mode=0\r\n
zfc_pairs=564,327;1120,213;\r\n
...
```

SET:

[http://camera\\_ip/hwlayer/extopt?zoom\\_pc=10&focus\\_pc=20&iris\\_pc=30&zfcovement=0&filter=1&save&wfilter=1](http://camera_ip/hwlayer/extopt?zoom_pc=10&focus_pc=20&iris_pc=30&zfcovement=0&filter=1&save&wfilter=1)  
[http://camera\\_ip/control/focuscontrol?zfc\\_addpair=1120,213&save&wfilter=1](http://camera_ip/control/focuscontrol?zfc_addpair=1120,213&save&wfilter=1)  
[http://camera\\_ip/control/focuscontrol?zfc\\_rmpair=1120,213&save&wfilter=1](http://camera_ip/control/focuscontrol?zfc_rmpair=1120,213&save&wfilter=1)  
[http://camera\\_ip/control/focuscontrol?zfc\\_mode=2&save&wfilter=1](http://camera_ip/control/focuscontrol?zfc_mode=2&save&wfilter=1)

### Note

- extopt/zfcovement (can be 0,1) is equal to focuscontrol/zfc\_mode, recommended to use zfc\_mode (can be 0,1,2)

## 2.11. FLASH CONTROL

**Module:** hwlayer/cperiph, config/ledpc**Manual:** Ch. 6.7 Flash Control**Get current settings**

GET:

[http://camera\\_ip/config/ledpc?getall&wfilter=1](http://camera_ip/config/ledpc?getall&wfilter=1)[http://camera\\_ip/hwlayer/cperiph?getall&wfilter=1](http://camera_ip/hwlayer/cperiph?getall&wfilter=1)

ANSWER:

```
...
mimetype=text/plain\r\n
sections=default,panel_0,panel_1,panel_2\r\n
npanels=3\r\n
...
```

[http://camera\\_ip/config/ledpc?getall&panel=0&wfilter=1](http://camera_ip/config/ledpc?getall&panel=0&wfilter=1)**Note**

- Index of the internal flash index: panel 0; external illuminators: panel 1, 2, ...  
npanels = 3, in this case 2 external illuminators are connected

ANSWER:

```
...
mimetype=text/plain\r\n
led_diff_pc=-1\r\n
led_pcdose=0\r\n
...
```

SET:

[http://camera\\_ip/config/ledpc?led\\_diff\\_pc=20&led\\_pcdose=10&panel=0&save&wfilter=1](http://camera_ip/config/ledpc?led_diff_pc=20&led_pcdose=10&panel=0&save&wfilter=1)[http://camera\\_ip/config/ledpc?led\\_diff\\_pc=-1&led\\_pcdose=0&panel=0&save&wfilter=1](http://camera_ip/config/ledpc?led_diff_pc=-1&led_pcdose=0&panel=0&save&wfilter=1)[http://camera\\_ip/hwlayer/cperiph?flashpwr=2&ftime\\_even=200&panel=0&ledena=1&wfilter=1](http://camera_ip/hwlayer/cperiph?flashpwr=2&ftime_even=200&panel=0&ledena=1&wfilter=1)[http://camera\\_ip/hwlayer/cperiph?ftime\\_odd=100&panel=0&fpfena=1&wfilter=1](http://camera_ip/hwlayer/cperiph?ftime_odd=100&panel=0&fpfena=1&wfilter=1)[http://camera\\_ip/hwlayer/cperiph?ledena=0&fpfena=0&panel=0&save&wfilter=1](http://camera_ip/hwlayer/cperiph?ledena=0&fpfena=0&panel=0&save&wfilter=1)**Note**

- In case of cameras equipped with rolling shutter sensors the camera illumination can be only enabled or disabled. For enabling: led\_pcdose=100

## 2.12. EVENT MANAGER

**Module:** trigger/eventman

**Manual:** Ch. 7.1.1 Event Manager

### Get available trigger sources

GET:

[http://camera\\_ip/trigger/eventman?getsources&wfilter=1](http://camera_ip/trigger/eventman?getsources&wfilter=1)

ANSWER:

```
...
mimetype=text/plain\r\n
sources=__motdet[3] (Hardware Motion
Detection), trigger/swtrigger[1] (Software
Trigger), trigger/uarttrigger[1] (UART Trigger), trigger/gpiotrigg[1] (GPIO
Trigger), trigger/scheduler[2] (Scheduler Trigger), trigger/vvq (Plate
Finder), trigger/radtrigger (Radar Trigger)\r\n
...
```

General syntax

```
sources=groupname/modulename [numberofoutputs] (stringname), groupname/modul
ename [numberofoutputs] (stringname), ... \r\n
```

### Get registered trigger sources and input configuration

[http://camera\\_ip/trigger/eventman?getconfig&wfilter=1](http://camera_ip/trigger/eventman?getconfig&wfilter=1)

ANSWER:

```
...
mimetype=text/plain\r\n
config=trigger/swtrigger/0:A, __motdet/0:B, trigger/uarttrigger/0:C\r\n
...
```

GENERAL SYNTAX:

```
config=
groupname/modulename/outputindex:assignedletter, groupname/modulename/outp
utindex:assignedletter\r\n
```

### Get/set formula

[http://camera\\_ip/trigger/eventman?getformula&wfilter=1](http://camera_ip/trigger/eventman?getformula&wfilter=1)

ANSWER:

```
...
mimetype=text/plain\r\n
formula=C\r\n
...
```

### Get/set state

[http://camera\\_ip/trigger/eventman?getstate&wfilter=1](http://camera_ip/trigger/eventman?getstate&wfilter=1)

ANSWER:

```
...  
mimetype=text/plain\r\n  
state=1\r\n  
...
```

SET:

Register

[http://camera\\_ip/trigger/eventman?register=\\_\\_motdet/0:A&wfilter=1](http://camera_ip/trigger/eventman?register=__motdet/0:A&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=\\_\\_motdet/1:B&wfilter=1](http://camera_ip/trigger/eventman?register=__motdet/1:B&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=trigger/swtrigger/0:C&wfilter=1](http://camera_ip/trigger/eventman?register=trigger/swtrigger/0:C&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=trigger/uarttrigger/0:D&wfilter=1](http://camera_ip/trigger/eventman?register=trigger/uarttrigger/0:D&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=trigger/gpiottrigger/0:E&wfilter=1](http://camera_ip/trigger/eventman?register=trigger/gpiottrigger/0:E&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=trigger/scheduler/0:F&wfilter=1](http://camera_ip/trigger/eventman?register=trigger/scheduler/0:F&wfilter=1)

[http://camera\\_ip/trigger/eventman?register=trigger/scheduler/1:G&wfilter=1](http://camera_ip/trigger/eventman?register=trigger/scheduler/1:G&wfilter=1)

#### Note

- The digit ID of each trigger source must be unique during the registration (recommended alphabetically, see above). The system will manage it automatically during registration via web interface.

### Start module

[http://camera\\_ip/trigger/eventman?formula=B&state=1&save&wfilter=1](http://camera_ip/trigger/eventman?formula=B&state=1&save&wfilter=1)

### Stop module

[http://camera\\_ip/trigger/eventman?state=0&save&wfilter=1](http://camera_ip/trigger/eventman?state=0&save&wfilter=1)

### Reset module

[http://camera\\_ip/trigger/eventman?state=2&save&wfilter=1](http://camera_ip/trigger/eventman?state=2&save&wfilter=1)

## 2.13. MOTION DETECTION TRIGGER

**Module:** capture/scapture

**Manual:** Ch. 7.1.2 Motion Detector

Get/set motion detection parameters

GET:

[http://camera\\_ip/capture/scapture?queryall&wfilter=X](http://camera_ip/capture/scapture?queryall&wfilter=X)

[http://camera\\_ip/capture/scapture?getall&wfilter=X&follow=0&frame=2](http://camera_ip/capture/scapture?getall&wfilter=X&follow=0&frame=2)

ANSWER:

```
...
mdsensitivity=75\r\n
mdspeed=75\r\n
mdreslevel=50\r\n
mdtriglevel=50,50\r\n
```

SET:

[http://camera\\_ip/capture/scapture?mdsensitivity=80&mdspeed=80&mdreslevel=55&mdtriglevel=55,55&wfilter=X&follow=0&frame=0](http://camera_ip/capture/scapture?mdsensitivity=80&mdspeed=80&mdreslevel=55&mdtriglevel=55,55&wfilter=X&follow=0&frame=0)

SAVE:

[http://camera\\_ip/setup/profiles?wfilter=X&save\\_motdets](http://camera_ip/setup/profiles?wfilter=X&save_motdets)

## 2.14. SOFTWARE TRIGGER

**Module:** trigger/swtrigger

**Manual:** Ch. 7.1.3 Software trigger

Get/set software trigger properties

GET:

[http://camera\\_ip/trigger/swtrigger?getall&output=0&wfilter=1](http://camera_ip/trigger/swtrigger?getall&output=0&wfilter=1)

ANSWER:

```
...
start_offs=0\r\n
end_offs=0\r\n
...
```

SET:

[http://camera\\_ip/trigger/swtrigger?sendtrigger=1&wfilter=1](http://camera_ip/trigger/swtrigger?sendtrigger=1&wfilter=1)

[http://camera\\_IP/trigger/swtrigger?start\\_offs=-100&end\\_offs=100&save&output=0&wfilter=1](http://camera_IP/trigger/swtrigger?start_offs=-100&end_offs=100&save&output=0&wfilter=1)

It is possible to attach max. 256 bytes of data to a software trigger (e.g. a trigger ID, device measurement data, etc.):

[http://camera\\_ip/trigger/swtrigger?sendtrigger=1&trigger\\_data=TRIGGER\\_0057&wfilter=1](http://camera_ip/trigger/swtrigger?sendtrigger=1&trigger_data=TRIGGER_0057&wfilter=1)

- This piece of data will be inserted in the uploaded data file (See 18. Upload Manager) with parameter name 'triggerdata'.
- In case of ANPR capable devices, trigger data will be available addressing the database column TRIGGER\_DATA (See 21. Database contents)

In both cases, trigger data will be encoded with Base64.



## 2.15. UART TRIGGER

**Module:** trigger/uarttrigger

**Manual:** Ch. 7.1.6 UART Trigger

Get/set UART trigger properties

GET:

[http://camera\\_ip/trigger/uarttrigger?getall&wfilter=1](http://camera_ip/trigger/uarttrigger?getall&wfilter=1)

[http://camera\\_ip/trigger/uarttrigger?getall&output=0&wfilter=1](http://camera_ip/trigger/uarttrigger?getall&output=0&wfilter=1)

SET:

[http://camera\\_ip/trigger/uarttrigger?baudrate=9600&bytesize=8&parity=0&stopbits=1&start\\_token=256&end\\_token=10&mode=0&save&wfilter=1](http://camera_ip/trigger/uarttrigger?baudrate=9600&bytesize=8&parity=0&stopbits=1&start_token=256&end_token=10&mode=0&save&wfilter=1)

[http://camera\\_ip/trigger/uarttrigger?start\\_offs=0&end\\_offs=0&mode=0&save&output=0&wfilter=1](http://camera_ip/trigger/uarttrigger?start_offs=0&end_offs=0&mode=0&save&output=0&wfilter=1)

### Note

- Start\_token=256 means: Start on first byte (decimal value)
- Start/End token can be decimal and hexadecimal (in case of hexadecimal the 0x prefix is needed)
- The website returns the values in decimal

## 2.16. RADAR TRIGGER (VIDAR WITH RADAR ONLY)

The appropriate setup cannot be completed with these requests only. The ANPR ROI's for the Radar trigger should be set via the web interface.

For more information, visit this site: [http://doc.arh.hu/cameras/Radar\\_Install\\_Guide\\_v1.0.pdf](http://doc.arh.hu/cameras/Radar_Install_Guide_v1.0.pdf)

GET:

[http://camera\\_ip/trigger/radtrigger?getstart\\_offs&getend\\_offs=0&getmode&output=0&wfilter=1](http://camera_ip/trigger/radtrigger?getstart_offs&getend_offs=0&getmode&output=0&wfilter=1)

[http://camera\\_ip/trigger/radtrigger?getadjl&getadjo&getdirection&section=radar&wfilter=](http://camera_ip/trigger/radtrigger?getadjl&getadjo&getdirection&section=radar&wfilter=)

### Note

To GET Direction:

- Approaching:  
<http://192.168.7.68/trigger/radtrigger?wfilter=1&section=radar&save&adjl=1572.0000&adjo=1296.0000&direction=1>
- Leaving:  
<http://192.168.7.68/trigger/radtrigger?wfilter=1&section=radar&save&adjl=1572.0000&adjo=1296.0000&direction=2>
- Approaching and Leaving:  
<http://192.168.7.68/trigger/radtrigger?wfilter=1&section=radar&save&adjl=1572.0000&adjo=1296.0000&direction=3>

SET:

[http://camera\\_ip/trigger/radtrigger?start\\_offs=0&end\\_offs=0&mode=0&output=0&save&wfilter=1](http://camera_ip/trigger/radtrigger?start_offs=0&end_offs=0&mode=0&output=0&save&wfilter=1)

[http://camera\\_ip/trigger/radtrigger?adjl=0&adjo=0&direction=0&section=radar&save&wfilter=1](http://camera_ip/trigger/radtrigger?adjl=0&adjo=0&direction=0&section=radar&save&wfilter=1)

## 2.17. GPIO TRIGGER

**Module:** trigger/gpiotrigger

**Manual:** Ch. 7.1.4 GPIO Trigger

Get/set GPIO trigger properties

GET:

[http://camera\\_ip/trigger/gpiotrigger?getall&wfilter=1](http://camera_ip/trigger/gpiotrigger?getall&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?getall&output=0&wfilter=1](http://camera_ip/trigger/gpiotrigger?getall&output=0&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?getgpin&wfilter=1](http://camera_ip/trigger/gpiotrigger?getgpin&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?getgpout&wfilter=1](http://camera_ip/trigger/gpiotrigger?getgpout&wfilter=1)

SET:

[http://camera\\_ip/trigger/gpiotrigger?gpout=1&wfilter=1](http://camera_ip/trigger/gpiotrigger?gpout=1&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?gpout=0&wfilter=1](http://camera_ip/trigger/gpiotrigger?gpout=0&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?gpout=0&samplerate=100&reqsamples=10&act\\_level=1&save&wfilter=1](http://camera_ip/trigger/gpiotrigger?gpout=0&samplerate=100&reqsamples=10&act_level=1&save&wfilter=1)

[http://camera\\_ip/trigger/gpiotrigger?start\\_offs=0&end\\_offs=0&mode=0&output=0&save&wfilter=1](http://camera_ip/trigger/gpiotrigger?start_offs=0&end_offs=0&mode=0&output=0&save&wfilter=1)

## 2.18. SCHEDULER TRIGGER

**Module:** trigger/scheduler

**Manual:** Ch. 7.1.5 Scheduler Trigger

Get/set scheduler trigger properties

GET:

[http://camera\\_ip/trigger/scheduler?getall&wfilter=1&output=0](http://camera_ip/trigger/scheduler?getall&wfilter=1&output=0)

[http://camera\\_ip/trigger/scheduler?getall&wfilter=1&output=1](http://camera_ip/trigger/scheduler?getall&wfilter=1&output=1)

SET:

[http://camera\\_ip/trigger/scheduler?enabled=1&mode=1&wday=mon,tue,wed,thu,fri,sat,sun;&day=every&hour=every&min=0;5;10;30-35;50;&sec=0;5;50-55;&output=0&save&wfilter=1](http://camera_ip/trigger/scheduler?enabled=1&mode=1&wday=mon,tue,wed,thu,fri,sat,sun;&day=every&hour=every&min=0;5;10;30-35;50;&sec=0;5;50-55;&output=0&save&wfilter=1)

[http://camera\\_ip/trigger/scheduler?enabled=0&output=0&save&wfilter=1](http://camera_ip/trigger/scheduler?enabled=0&output=0&save&wfilter=1)

## 2.19. PLATE FINDER

**Module:** trigger/vvq

**Manual:** Ch. 7.1.7 Plate Finder

SET: [http://camera\\_ip/trigger/vvq?roi=10,10,80,80&save&wfilter=1](http://camera_ip/trigger/vvq?roi=10,10,80,80&save&wfilter=1)

[http://camera\\_ip/trigger/vvq?enabled=0&save&wfilter=1](http://camera_ip/trigger/vvq?enabled=0&save&wfilter=1)

GET:

[http://camera\\_ip/trigger/vvq?getenabled=0&getroi&wfilter=1](http://camera_ip/trigger/vvq?getenabled=0&getroi&wfilter=1)

## 2.20. LASER TRIGGER

**Module:** trigger/dsttrigger

**Manual:** Ch. 7.1.8 Plate Finder Trigger

GET:

[http://camera\\_ip/trigger/dsttrigger?getall&wfilter=1](http://camera_ip/trigger/dsttrigger?getall&wfilter=1)

[http://camera\\_ip/trigger/dsttrigger?getall&device=0&wfilter=1](http://camera_ip/trigger/dsttrigger?getall&device=0&wfilter=1)

[http://camera\\_ip/trigger/dsttrigger?getT\\_health&device=0&wfilter=1](http://camera_ip/trigger/dsttrigger?getT_health&device=0&wfilter=1)

## 2.21. UPLOAD MANAGER (NON-SMART CAMERAS ONLY)

**Module:** upload/uploadman

**Manual:** Ch. 7.2.14 Result Upload

Get log of event upload attempts

GET:

[http://camera\\_ip/upload/uploadman?getlog](http://camera_ip/upload/uploadman?getlog)

ANSWER:

```
...
2,15-13/2221093.jpg,0,1,Couldn't resolve host name;0,15-
13/2331476.jpg,0,0,No error;0x00
...
```

GENERAL SYNTAX:

```
protocol,filenametemplate,include,encode,estring;...;0x00
```

```
protocol: [ftp:0, smtp:1, http:2, gxdctrf:3, invalid:4]
```

```
content: [image+data: 0, image: 1, data: 2]
```

```
encode: [0: OK, 1: failed, 2: image OK, data failed]
```

### Note

- If no log information exists, 1 byte of data will be returned (0x00)
- At the end there will be one 0x00 byte even if there is log information.
- 2.27-es pont a hivatkozás

Upload data to FTP server

GET:

[http://camera\\_ip/upload/uploadman?getall&wfilter=X](http://camera_ip/upload/uploadman?getall&wfilter=X)

ANSWER:

```
...
sections=default,protocol_smtp,protocol_ftp,protocol_post\r\n
currentprotocol=0\r\n
...
```

[http://camera\\_ip/upload/uploadman?getall&wfilter=X&protocol=ftp](http://camera_ip/upload/uploadman?getall&wfilter=X&protocol=ftp)

[http://camera\\_ip/upload/uploadman?content=0&currentprotocol=0&host=testserver&username=&password=&filenametemplate=\\$d-\\$h/\\$m\\$S\\$I&protocol=ftp&save&wfilter=X](http://camera_ip/upload/uploadman?content=0&currentprotocol=0&host=testserver&username=&password=&filenametemplate=$d-$h/$m$S$I&protocol=ftp&save&wfilter=X)

Upload Data to SMTP Server (e-mail)

GET:

[http://camera\\_ip/upload/uploadman?getall&wfilter=X](http://camera_ip/upload/uploadman?getall&wfilter=X)

ANSWER:

```
...  
sections=default,protocol_smtp,protocol_ftp,protocol_post\r\n  
currentprotocol=0\r\n  
...
```

```
http://camera\_ip/upload/uploadman?getall&wfilter=X&protocol=smtp  
http://camera\_ip/upload/uploadman?content=0&currentprotocol=1&host=testserver&username=&password=&from=&to=user1@testserver;user2@testserver&filenameemplate=\$d-\$h/\$m\$s\$I&protocol=smtp&save&wfilter=X
```

### Upload Data to HTTP Server

GET:

```
http://camera\_ip/upload/uploadman?getall&wfilter=X
```

ANSWER:

```
...  
sections=default,protocol_smtp,protocol_ftp,protocol_post\r\n  
currentprotocol=0\r\n  
...
```

```
http://camera\_ip/upload/uploadman?getall&wfilter=X&protocol=post  
http://camera\_ip/upload/uploadman?content=0&currentprotocol=2&host=testserver&filenameemplate=\$d-\$h/\$m\$s\$I&protocol=post&save&wfilter=X
```



## 2.22. SET/GET ANPR PROPERTIES (SMART CAMERAS ONLY)

**Module:** lpr/cff**Manual:** Ch. 7.2.1 Browse**Note**

This module will deliver its response in the binary part either as an XML document or a JPEG file.

**Get a property of the License Plate Recognition module**

GET:

[http://\[camera\\_ip\]/lpr/cff?cmd=getproperty&name=\[property\\_path\]](http://[camera_ip]/lpr/cff?cmd=getproperty&name=[property_path])

ANSWER:

```
...
<ans>\r\n
  <property>\r\n
    <[property name] value ="[property value]"/>\r\n
  </property/>\r\n
</ans>
...
```

**Set a property of the License Plate Recognition module**

GET:

[http://\[camera\\_ip\]/lpr/cff?cmd=setproperty&name=\[property\\_path\]&value=\[property value\]](http://[camera_ip]/lpr/cff?cmd=setproperty&name=[property_path]&value=[property value])

ANSWER:

```
...
<ans>\r\n
  <ecode value ="0"/>\r\n
</ans>
...
```

**Save properties of the License Plate Recognition module**

GET

[http://\[camera\\_ip\]/lpr/cff?cmd=saveproperties](http://[camera_ip]/lpr/cff?cmd=saveproperties)

ANSWER:

```
...
<ans>\r\n
<ecode value="0"/>\r\n
</ans>...
```

## 2.23. BROWSE THE DATABASE (SMART CAMERAS ONLY)

Module: lpr/cff

Manual: Ch. 7.2.1 Browse

**Note**

This module will deliver its response in the binary part either as an XML document or a JPEG file.

**Get last record ID**

GET:

[http://camera\\_ip/lpr/cff?cmd=getid&id=last](http://camera_ip/lpr/cff?cmd=getid&id=last)

ANSWER:

```
...
<ans>\r\n
<ecode value="0"/>\r\n
<id value="1396268058242"/>\r\n
</ans>
```

...

**Get data file corresponding to ID**

GET:

[http://camera\\_ip/lpr/cff?cmd=getdata&id=last](http://camera_ip/lpr/cff?cmd=getdata&id=last)

[http://camera\\_ip/lpr/cff?cmd=getdata&id=1396268058242](http://camera_ip/lpr/cff?cmd=getdata&id=1396268058242)

ANSWER:

```
...
<result>
  <location value="-"/>
  <cameraid value="00001009"/>
  <ID value="1396306892338"/>
  <image_hash value="D834B424BBE513390F635AD59E8EDED796D978F0"/>
  <capture>
    <frametime value="2014.04.02 3:59:16.733"/>
    <frametimems value="1396403956733"/>
    <frameindex value="0"/>
  </capture>
  <anpr>
    <text value="ARH002"/>
    <type value="111"/>
    <frame value="578,443,734,412,730,445,574,476"/>
    <bgcolor value="16777215"/>
    <color value="0"/>
    <confidence value="94"/>
    <timems value="360"/>
    <resultcnt value="1"/>
  </anpr>
</motdet>
```

```
<rect value="0,0,1,1"/>
<confidence value="100"/>
<objectid value="22257"/>
<objectix value="4"/>
</motdet>
<trigger>
  <speed value="52.00"/>
  <speed_limit value="1.00"/>
  <direction value="1"/>
  <category value="679"/>
  <vclass value="1"/>
  <timems value="0"/>
</trigger>
<misc>
  <gps_lat value="47.4930"/>
  <gps_lon value="19.0234"/>
</misc>
</result>
```

### Get image corresponding to ID

GET:

[http://camera\\_ip/lpr/cff?cmd=getimage&id=last](http://camera_ip/lpr/cff?cmd=getimage&id=last)

[http://camera\\_ip/lpr/cff?cmd=getimage&id=1396268058242](http://camera_ip/lpr/cff?cmd=getimage&id=1396268058242)

[http://camera\\_ip/lpr/cff?cmd=getimage&id=1396268058242](http://camera_ip/lpr/cff?cmd=getimage&id=1396268058242) (cropped license plate image, will not exist, if plate was not found)





## 2.24. DATABASE CONTENTS (SMART CAMERAS ONLY)

The following table lists the data stored to each event

Field	Type	Description
ADR_BGCOLOR	INTEGER	ADR plate background color
ADR_COLOR	INTEGER	ADR plate text color
ADR_CONFIDENCE	INTEGER	ADR plate confidence
ADR_FRAME	TEXT	ADR plate coordinates (x0,y0,...,x3,y3)
ADR_FRAME_X1	INTEGER	ADR frame coordinate x1
ADR_FRAME_X2	INTEGER	ADR frame coordinate x2
ADR_FRAME_X3	INTEGER	ADR frame coordinate x3
ADR_FRAME_X4	INTEGER	ADR frame coordinate x4
ADR_FRAME_Y1	INTEGER	ADR frame coordinate y1
ADR_FRAME_Y2	INTEGER	ADR frame coordinate y2
ADR_FRAME_Y3	INTEGER	ADR frame coordinate y3
ADR_FRAME_Y4	INTEGER	ADR frame coordinate y4
ADR_TEXT	TEXT	ADR plate text
ADR_TYPE	INTEGER	ADR plate type
ANPR_BGCOLOR	INTEGER	License plate background color
ANPR_CATEGORY	TEXT	License plate category (platetypeconf/platetypeinfos/category)
ANPR_COLOR	INTEGER	License plate text color
ANPR_CONFIDENCE	INTEGER	License plate confidence
ANPR_FRAME	TEXT	License plate coordinates (x0,y0,...,x3,y3)
ANPR_FRAME_X1	INTEGER	LP frame coordinate x1
ANPR_FRAME_X2	INTEGER	LP frame coordinate x2

Field	Type	Description
ANPR_FRAME_X3	INTEGER	LP frame coordinate x3
ANPR_FRAME_X4	INTEGER	LP frame coordinate x4
ANPR_FRAME_Y1	INTEGER	LP frame coordinate y1
ANPR_FRAME_Y2	INTEGER	LP frame coordinate y2
ANPR_FRAME_Y3	INTEGER	LP frame coordinate y3
ANPR_FRAME_Y4	INTEGER	LP frame coordinate y4
ANPR_PLATESRC	INTEGER	ANPR source (0 : none, 1 : main, 2 : overview image)
ANPR_RESULTCNT	INTEGER	Number of identical readings
ANPR_TEXT	TEXT	License plate text
ANPR_TIMEMS	INTEGER	Recognition time [ms]
ANPR_TYPE	INTEGER	License plate jurisdiction code
AUXIMG_HASH	TEXT	Overview image hash (SHA1)
BELTIMG_HASH	TEXT	Belt image hash (SHA1)
CMHWAY_BELT_CONF	REAL	Seatbelt detected, confidence
CMHWAY_OPT_SPEED	REAL	Optical speed [km/h]
CMHWAY_OPT_SPEED_CONF	REAL	Optical speed confidence
COUNTRY_LONG	TEXT	License plate jurisdiction/country, textual, long
COUNTRY_SHORT	TEXT	License plate jurisdiction/country, textual, short
CTRL_AGAIN	UNSIGNED SMALLINT	Image analog gain
CTRL_BLACKLEVEL	UNSIGNED SMALLINT	Image blacklevel
CTRL_DGAIN	UNSIGNED SMALLINT	Image digital gain
CTRL_IRIS	UNSIGNED SMALLINT	Image iris

Field	Type	Description
CTRL_SHUTTERMS	UNSIGNED INTEGER	Image shutter [us]
EVENT_LOG	TEXT	Processing log (base64)
EVTS_DATA_ACK_TIME	BIGINT	Deprecated
EVTS_DATA_EXTID	TEXT	Deprecated
EVTS_DATA_IS_ACK	UNSIGNED SMALLINT	Deprecated
EVTS_DATA_IS_FINISHED	UNSIGNED SMALLINT	Deprecated
EVTS_DATA_IS_SENT	UNSIGNED SMALLINT	Deprecated
EVTS_DATA_SEND_TIME	BIGINT	Deprecated
EVTS_DATA_TRYING	UNSIGNED SMALLINT	Deprecated
EVTS_IMG0_ACK_TIME	BIGINT	Deprecated
EVTS_IMG0_EXTID	TEXT	Deprecated
EVTS_IMG0_IS_ACK	UNSIGNED SMALLINT	Deprecated
EVTS_IMG0_IS_FINISHED	UNSIGNED SMALLINT	Deprecated
EVTS_IMG0_IS_SENT	UNSIGNED SMALLINT	Deprecated
EVTS_IMG0_SEND_TIME	BIGINT	Deprecated
EVTS_IMG0_TRYING	UNSIGNED SMALLINT	Deprecated
FRAMEINDEX	UNSIGNED INTEGER	Image frame index
FRAMETIMEMS	BIGINT	Image /event timestamp [ms]
HASH	TEXT	Main image hash (SHA1)
HYD_CONF	INTEGER	HyDetect confidence
HYD_FRAME_X1	INTEGER	HyDetect vehicle bounding rectangle x1
HYD_FRAME_X2	INTEGER	HyDetect vehicle bounding rectangle x2

Field	Type	Description
HYD_FRAME_X3	INTEGER	HyDetect vehicle bounding rectangle x3
HYD_FRAME_X4	INTEGER	HyDetect vehicle bounding rectangle x4
HYD_FRAME_Y1	INTEGER	HyDetect vehicle bounding rectangle y1
HYD_FRAME_Y2	INTEGER	HyDetect vehicle bounding rectangle y2
HYD_FRAME_Y3	INTEGER	HyDetect vehicle bounding rectangle y3
HYD_FRAME_Y4	INTEGER	HyDetect vehicle bounding rectangle y4
HYD_ISVEH	INTEGER	HyDetect vehicle present (1 : yes, 0 : no)
ID	BIGINT	Primary key/Event ID
IMGSIZ_AUX_X	INTEGER	Overview image x size [px]
IMGSIZ_AUX_Y	INTEGER	Overview image y size [px]
IMGSIZ_BELT_X	INTEGER	Belt image x size [px]
IMGSIZ_BELT_Y	INTEGER	Belt image y size [px]
IMGSIZ_LP_X	INTEGER	Plate image x size [px]
IMGSIZ_LP_Y	INTEGER	Plate image y size [px]
IMGSIZ_NORM_X	INTEGER	Main image x size [px]
IMGSIZ_NORM_Y	INTEGER	Main image y size [px]
IMGSIZ_STP_X	INTEGER	Strip image x size [px]
IMGSIZ_STP_Y	INTEGER	Strip image y size [px]
LP_BRG	INTEGER	Deprecated
LP_SRP	INTEGER	Deprecated
LPIMG_HASH	TEXT	Plate image hash (SHA1)
MD_CONFIDENCE	INTEGER	Motion detection confidence

Field	Type	Description
MD_FRAME_BOTTOM	INTEGER	Motion detection frame bottom
MD_FRAME_LEFT	INTEGER	Motion detection frame left
MD_FRAME_RIGHT	INTEGER	Motion detection frame right
MD_FRAME_TOP	INTEGER	Motion detection frame top
MD_OBJECTID	UNSIGNED INTEGER	Motion detection Event ID
MD_OBJECTIX	UNSIGNED INTEGER	Motion detection Image index (nth frame of the event)
MD_RECT	TEXT	Motion frame (left,top,right,bottom)
MISC_GPS_LAT	TEXT	GPS latitude
MISC_GPS_LON	TEXT	GPS longitude
MMR_CATEGORY	TEXT	MMR category
MMR_CATEGORY_CONF	INTEGER	MMR category confidence
MMR_COLOR	TEXT	MMR color
MMR_COLOR_CONF	INTEGER	MMR color confidence
MMR_MAKE	TEXT	MMR make
MMR_MODEL	TEXT	MMR model
MMR_MODEL_CONF	INTEGER	MMR model confidence
MMR_SUBMODEL	TEXT	MMR submodel
STATE_LONG	TEXT	License plate jurisdiction/state, textual, long
STATE_SHORT	TEXT	License plate jurisdiction/state, textual, short
STRIPIMG_HASH	TEXT	Strip image hash (SHA1)
TRIGGER_CATEGORY	INTEGER	Vehicle e-length
TRIGGER_DATA	BLOB	Trigger data (base64)

Field	Type	Description
TRIGGER_DATALEN	INTEGER	Trigger data length
TRIGGER_DIRECTION	SMALLINT	Vehicle heading (1 : approaching, 2 : leaving)
TRIGGER_SOURCE	TEXT	Trigger source
TRIGGER_SPEED	UNSIGNED INT	Vehicle speed
TRIGGER_SPEEDLIMIT	INTEGER	Vehicle speed limit
TRIGGER_TIMEMS	BIGINT	Trigger time [ms]
TRIGGER_VCLASS	INTEGER	Vehicle class (user defined limits)



## 2.25. QUERY DATABASE (SMART CAMERAS ONLY)

Module: lpr/cff

Manual: Ch. 7.2.1 Browse

**Note**

This module will deliver its response in the binary part either as an XML document or a JPEG file.

**Steps of retrieving results from the database:**

1. Query IDs with regular SQL commands formulating searching criteria
2. Get image and data of the events corresponding to the returned IDs  
(see Browse the database Programmers section)

**Executing an SQL query**

GET:

[http://camera\\_ip/lpr/cff?cmd=getresultlist&select=WHERE ID>0](http://camera_ip/lpr/cff?cmd=getresultlist&select=WHERE ID>0)

ANSWER:

```
<results>
<result_0 value="1633966169774"/>
<result_1 value="1633966169773"/>
<result_2 value="1633966169772"/>
<result_3 value="1633966169771"/>
.
.
.
<result_396 value="1633966169378"/>
<result_397 value="1633966169377"/>
<n_results value="398"/>
</results>
```

**Note**

- default value of max\_record parameter is 1000. This value can be increased using the default/cfs/db/max\_record parameter

GET:

[http://camera\\_ip/lpr/cff?cmd=getresultlist&select=WHERE ID>1396465612672 AND ID<1396465612680](http://camera_ip/lpr/cff?cmd=getresultlist&select=WHERE ID>1396465612672 AND ID<1396465612680)

[http://camera\\_ip/lpr/cff?cmd=getresultlist&select=WHERE ANPR\\_TEXT LIKE '%5%'](http://camera_ip/lpr/cff?cmd=getresultlist&select=WHERE ANPR_TEXT LIKE '%5%')

**Note**

- use '%25' instead of '%' in the web browser  
ex.: ... LIKE '%255%25'

GET:

[http://camera\\_ip/lpr/cff?cmd=getresultlist&select=WHERE TRIGGER\\_SPEED>5000](http://camera_ip/lpr/cff?cmd=getresultlist&select=WHERE TRIGGER_SPEED>5000)**Note**

- 'speed' and 'speed limit' values are stored as fixed point numbers (multiplied by 100), thus SQL queries must be formulated accordingly: 50.00 km/h should be entered as 5000

Last Event ID:

[http://camera\\_ip/lpr/cff?cmd=getid&id=last](http://camera_ip/lpr/cff?cmd=getid&id=last)

ANSWER:

```
<ans>
<rcode value="0"/>
<id value="1633966170027"/>
</ans>
```

Details of the last event:

[http://camera\\_ip/lpr/cff?cmd=getdata&id=last](http://camera_ip/lpr/cff?cmd=getdata&id=last)

ANSWER:

```
<result>
<location value="Test location"/>
<cameravid value="2200888"/>
<ID value="1633966169894"/>
<image_hash value="09147619822f582f10c75b1ae6b9b2baf12b2e61"/>
<capture>
<frametime value="2021.10.11 15:47:01.260"/>
<frametimems value="1633967221260"/>
<frameindex value="157191"/>
</capture>
<anpr>
<text value="RIU701"/>
<type value="101011"/>
<country value="HUN"/>
.
.
.
```



Details of an event with a specific ID:

[http://camera\\_ip/lpr/cff?cmd=getdata&id=1633966169378](http://camera_ip/lpr/cff?cmd=getdata&id=1633966169378)

ANSWER:

```
<result>
<location value="Test location"/>
<cameraid value="2200888"/>
<ID value="1633966169378"/>
<image_hash value="3da8434d18fce551c7fd3dc05948a34db5932711"/>
<capture>
<frametime value="2021.10.11 15:28:46.995"/>
<frametimems value="1633966126995"/>
<frameindex value="135305"/>
</capture>
<anpr>
<text value="n.a."/>
<type value="0"/>
<country value=""/>
.
.
.
```

Image for an event with a given ID:

[http://camera\\_ip/lpr/cff?cmd=getimage&id=1633966169378](http://camera_ip/lpr/cff?cmd=getimage&id=1633966169378)

ANSWER:

```
<picture>
```

To retrieve the ID of events where the license plate contains the string "MB":

[http://camera\\_ip/lpr/cff?cmd=querydb&sql=select id from cffresult where anpr\\_text like '%MB%'](http://camera_ip/lpr/cff?cmd=querydb&sql=select id from cffresult where anpr_text like '%MB%')

ANSWER:

```
<results>
<row>
<ID value="1633966169787"/>
</row>
<row>
<ID value="1633966170001"/>
</row>
<row>
<ID value="1633966170297"/>
</row>
<n_results value="3"/>
</results>
```

Query the license plates of an event where the trigger speed is greater than 50:

[http://camera\\_ip/lpr/cff?cmd=querydb&sql=select anpr\\_text from cffresult where trigger\\_speed > 50](http://camera_ip/lpr/cff?cmd=querydb&sql=select anpr_text from cffresult where trigger_speed > 50)

ANSWER:

```
<results>
<row>
<ANPR_TEXT value="ARH001"/>
</row>
<row>
<ANPR_TEXT value="ABC123"/>
</row>
<n_results value="2"/>
</results>
```

Query the ID and FRAMETIMEMS value of events where the event created between 2021-10-11 16:35:00 and 2021-10-11 16:35:10:

[http://camera\\_ip/lpr/cff?cmd=querydb&sql=SELECT id, frametimems from cffresult where substr\(frametimems,1,10\) <TIME\\_COMPENSATION\\_SEC> BETWEEN cast\(strftime\('%s','2021-10-11 16:35:00'\) as int\) and cast\(strftime\('%s','2021-10-11 16:35:10'\) as int\)](http://camera_ip/lpr/cff?cmd=querydb&sql=SELECT id, frametimems from cffresult where substr(frametimems,1,10) <TIME_COMPENSATION_SEC> BETWEEN cast(strftime('%s','2021-10-11 16:35:00') as int) and cast(strftime('%s','2021-10-11 16:35:10') as int))

#### Note

The value of <TIME\_COMPENSATION\_SEC> depends on which time zone the user is. In the ANPR database, the time of events is stored in UTC and the time of the events to be filtered during the query must be shifted according to the geographically valid time zone for the time stored in UTC. For example, in the Europe / Budapest time zone in summertime will be +1 hour plus compared to UTC, so <TIME\_COMPENSATION\_SEC> will be +3600. The '+' sign with URL encoding is '% 2b' and the '-' sign is '% 2d'.

ANSWER:

```
<results>
<row>
<ID value="1633966169571"/>
<FRAMETIMEMS value="1633966503733"/>
</row>
<row>
<ID value="1633966169572"/>
<FRAMETIMEMS value="1633966505533"/>
</row>
<row>
<ID value="1633966169573"/>
<FRAMETIMEMS value="1633966508033"/>
</row>
<n_results value="3"/>
</results>
```

## 2.26. TEMPLATES (SMART CAMERAS ONLY)

The content of the result data file (and the file name) is customizable using templates. The following short example, which will describe a result formatted as an XML contains all features:

```

1      <?xml version="1.0" encoding="UTF-8"?>
2      <result>
3          <ID value ="$(ID) ;"/>
4          <text value="$DB2XML($(ANPR_TEXT)) ;"/>
5          <location value="$(location) ;"/>
6          <image value="$(normal_img) ;"/>
7      </result>

```

Lines #1, #2 and #7 are text only (of course in this example we have to adhere to the XML standard). Line #3 is an example of a database field, line #4 is an example of a function, line 5 and 6 are examples of a property/special field. This template may evaluate to something like:

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
<ID value ="123456789"/>
<text value="ABC123"/>
<location value="Test Site 34b"/>
<image value="/9j/4AAQSkZJRgABAQAAQABA[... a base 64 encoded
image...]"/>
</result>

```

The same content in a different format might be:

```

1      event_id=$(ID) ;
3      plate_text=$DB2XML($(ANPR_TEXT)) ;
4      camera_location=$(location) ;
5      vehicle_image=$(normal_img) ;

```

which will evaluate to:

```
event_id=123456789
plate_text=ABC123
camera_location=Test Site 34b
vehicle_image=/9j/4AAQSkZJRgABAQAAAQABA[... a base 64 encoded image...]
```

### Database fields

The contents of the database (see Chapter ...) can be injected using the following syntax:  
`$([database field id])`

### Functions

To format the output the following functions are provided, with the general syntax:  
`$(function name)([argument1],[argument2],..., [argumentN]);`

Plate text formatting:

**DB2XML**(*text, flags: optional*) - converts the license plate text *text* from DB to XML format

**DB2JSON**(*text, flags: optional*) - converts the license plate text *text* from DB to JSON format

**DB2UTF8**(*text, flags: optional*) - converts the license plate text *text* from DB to UTF8

The following flags are available:

ARABIC2LATIN - which converts arabic numbers to their latin equivalent, and non-numeric arabic to ' '.

Time formatting:

**FormatTime**(*timestamp, format\_string: optional*) - formats timestamp given in milliseconds according to the optional format\_string (see the POSIX strftime function) For example:

```
"$FormatTime($ (FRAMETIMEMS), %Y%m%dT%H%M%S%z) "
```

evaluates to "20180419T145713+0200". If the format string is omitted, the same timestamp is evaluated as "2018.04.19 14:57:13.594".

### Special fields

The following keywords are defined:

`normal_img` - the image representing the event, Base64 encoded

`strip_img` - image strip Base64 encoded

`lp_img` - the cropped license plate image, Base64 encoded encoded < imagestrip value="\$(strip\_img);"/ >

`aux_img` - the overview image, Base64 encoded

`location` - the location string as defined in default/cfs/db/location

`cameraid` - the camera HW id, may be overridden with default/cfs/db/cameraid

**Note****Special characters**

The character '\$' must always be escaped with '\'. Within an expression the characters '\$', '(', ')', ';' and ',' have to be escaped with a '\' character. For example the following expression:

```
() ; $TEST (a\$a\, , $ (LP) ) ;
```

will evaluate as

```
() ; a$a, DATA
```

provided that the function TEST concatenates its arguments and the field LP contains the text 'DATA'.

**Administration**

You can upload a template with the `cmd = uploadtemplate` command as an HTTP POST request as multipart / form-data. As a parameter, enter the name you want to reference the template in the `updatefilename` field.

The device template in use and the system template cannot be deleted / overwritten.

The browser sends with this header:

```
Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

Connection: keep-alive

Content-Length: 889

Content-Type: multipart/form-data; boundary=----
WebKitFormBoundaryCAHOjxAq53NgEqQK

Host: 121d219.ar.local

Origin: http://121d219.ar.local

Referer: http://121d219.ar.local/

User-Agent: Mozilla/5.0 (X11; Fedora; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107
Safari/537.36
```

Form Data:

```
-----WebKitFormBoundaryCAHOjxAq53NgEqQK

Content-Disposition: form-data; name="updatefilename";
filename="proba.tpl"

Content-Type: application/octet-stream

-----WebKitFormBoundaryCAHOjxAq53NgEqQK--
```

The list of available templates can be queried with the command `cmd = listtemplates`, the device will give an XML response like this:

```
<ans>
<template0 value="file_default.sys.tpl"/>
<template1 value="file_json_default.sys.tpl"/>
<template2 value="json_cmgo.sys.tpl"/>
<template3 value="json_default.sys.tpl"/>
<template4 value="xml_data.sys.tpl"/>
<template5 value="xml_default.sys.tpl"/>
<template6 value="xml_evts.sys.tpl"/>
<template7 value="xml_minimal.sys.tpl"/>
</ans>
```

Templates can be deleted with the `cmd = removetemplate` command (system templates are not allowed to be deleted), e.g. `lpr/cff?cmd=removetemplate&name=proba.tpl`.

The device generates the output files using the templates specified in the properties under the `/ default / cfs / template` group. The template name want to be used must be entered here.

**template** - formats the web search (getdata) with this template

**user\_template** - formats the files to be uploaded with this template

**filename\_template** - format the names of the files to be uploaded with this template

## 2.27. ENGINE MANAGER (SMART CAMERAS ONLY)

Module: lpr/cff

Manual: Ch. 7.2.4 Engine Manager

### Note

This module will deliver its response in the binary part either as an XML document or a JPEG file.

### Pipeline

The pipeline can be built freely, up to 8 elements, in any order. The following example can be configured.

```
<structure>
  <stage0>
    <type value="scapture"/>
    <grp value="capture"/>
  </stage0>
  <stage1>
    <type value="presel"/>
    <grp value="presel"/>
  </stage1>
  <stage2>
    <type value="anpr_main"/>
    <grp value="recognize"/>
  </stage2>
  <stage3>
    <type value="post"/>
    <grp value="postproc"/>
  </stage3>
  <stage4>
    <type value="void"/>
    <grp value="void"/>
  </stage4>
  <stage5>
    <type value="void"/>
    <grp value="void"/>
  </stage5>
  <stage6>
    <type value="void"/>
    <grp value="void"/>
  </stage6>
  <stage7>
    <type value="void"/>
    <grp value="void"/>
  </stage7>
</structure>
```

The pipeline will be built from the elements in the specified order (0-7). Several of the same elements can be placed in the pipeline with different settings, in which case it is advisable to use a separate path to store the properties (see below). Note that adding any extra stages (especially analytics) will seriously increase processing time.

The type of stage must be specified in the type field, the following are possible:

*scapture* - image receiving stage, it is advisable to put at least one at the beginning of the processing  
*preasel* - preselector, gets all incoming images, runs ANPR (preferably on a fast vq / vvq engine) and segments the image sequence into passages based on the results  
*anpr\_main* - ANPR stage, this is where the type engine will run  
*cmhway* - search for optical speed and seat belt  
*mmr* - Make and Model Recognition  
*hyd* - vehicle detection  
*post* - post-processing (attaching thumbnails, captioning images, etc.)

Each stage has a property group where its settings are stored, this must be specified in the *grp* field. For example, above the *stage3* stage (post) settings are under the *default / cff / postproc* path.

## Stages

The following stages can be stitched in any order you like (most of the possible combinations are obviously meaningless, but this is the only way to ensure flexibility).



## Single Capture

Its task is to receive and decompress the images and provide them with an event ID and index.

```

<capture>
  <misc>
    <manual_direction value="0"/>
  </misc>
  <aux>
    <id value="203.0.113.1"/>
    <enabled value="0"/>
    <delays value="0"/>
  </aux>
  <filter>
    <direction value="0"/>
    <speeding_only value="0"/>
    <min_trg_pulse value="0"/>
  </filter>
  <manual_frame>
    <left value="0"/>
    <top value="0"/>
    <right value="46"/>
    <bottom value="83"/>
  </manual_frame>
  <classes>
    <class00>
      <speed_limit value="100"/>
      <range_min value="10"/>
      <range_max value="499"/>
      <name value="CAR"/>
    </class00>
    <class01>
      <speed_limit value="90"/>
      <range_min value="500"/>
      <range_max value="1299"/>
      <name value="TRUCK"/>
    </class01>
    <class02>
      <speed_limit value="80"/>
      <range_min value="1300"/>
      <range_max value="3999"/>
      <name value="LONG TRUCK"/>
    </class02>
    <class03>
      <speed_limit value="9999"/>
      <range_min value="0"/>
      <range_max value="9"/>
      <name value="OUT OF RANGE (S)"/>
    </class03>
    <class04>
      <speed_limit value="9999"/>
      <range_min value="4000"/>
      <range_max value="999999"/>
      <name value="OUT OF RANGE (L)"/>
    </class04>
  </classes>
  <image_strip>
    <enabled value="0"/>
    <timeout value="6000"/>
    <scale_factor value="33"/>
  </image_strip>
  <gps>
    <enabled value="0"/>
  </gps>
  <proc>
    <fifo_maxitems value="300"/>
    <priority value="-2"/>
    <port value="4444"/>
    <debug value="0"/>
  </proc>
</capture>

```

*misc / manual\_direction* - Determines the direction of traffic as follows:

- if you force it with the *\* manual\_direction* property, it will overwrite everything (see the negative values in the table below), otherwise
- if the trigger sends direction, it will be, otherwise
- if it does not send direction but sends speed, then if negative, then leaving, if positive, then approaching, otherwise
- if you do not send a speed, the value of **manual\_direction** will be (see the positive values in the table below), otherwise
- it is unknown.

<i>misc/manual_direction</i> value	function
0	does nothing
1	if no direction, ARRIVING
2	if there is no direction, LEAVING
-1	be sure it is ARRIVING
-2	be sure it is LEAVING

- *aux / id* - Overview camera IP (can't resolve URL)
- *aux / enabled* - Enable OV camera

<i>aux / enabled</i> value	function
0	none
1	overview only
2	If you do not find a license plate on the image, see the overview as well

- *aux / delayms* - fixed offset between normal and OV image, in milliseconds, requesting a later image as OV
- *filter / direction* - interesting for radar, directional filtering as for radar, 0 - no filtering
- *filter / speeding\_only* - interesting for radar, filtering according to the speed limits specified in classes, lower speed events are dropped, 0 – turned off
- *filter / min\_trg\_pulse* - 0 minimum e-length for radar, under it will be discard
- *manual\_frame / left* - manually specified frame in % of the image
- *manual\_frame / top* - manually specified frame in % of the image
- *manual\_frame / right* - manually specified frame in % of the image
- *manual\_frame / bottom* - manually specified frame in % of the image
- *classes / class00 / speed\_limit* - 5 classes can be specified, in which the passages can be classified according to e-length, and the speed limit for them
- *classes / class00 / range\_min* - the minimum of the class e-length
- *classes / class00 / range\_max* - the maximum of the class e-length
- *classes / class00 / name* - the name of the class (eg car, small truck, etc.)
- *image\_strip / enabled* - whether to create a series of thumbnails of the passages (help to set up a trigger). IMPORTANT! all images have to be compressed / reduced for this, so it will be very slow if we are in vvq / mx04 performance mode, it is advisable to turn it off!
- *image\_strip / timeout* - to close the event thumbnail timeout (in ms), if the trigger end signal does not appear, max. it waits for a given time before closing the thumbnail
- *image\_strip / scale\_factor* - reduce thumbnail (% of original image)
- *gps / enabled* - whether to read GPS data
- *proc / fifo\_maxitems* - storage size, will probably never need to be set
- *proc / priority* - line priority
- *proc / port* - the value of the image receiving port (this should also be used on the sending side if you change it)

## Preselection

type: *presel*

Its task is to quickly pre-select the images already grouped by the trigger into a pass, segmenting them into multiple passes if necessary, and selecting the image that best represents the pass, from which the more time-consuming processing steps will be performed later in the pipeline. (The terms "event" and "passage" are used interchangeably.) Of the similar license plates in the passage pictures, the one that best represents the passage should be selected. License plate position, syntax, etc. based on the pictures you get points and the license plate that gets the most points will be the selected picture.

```
<presel>
  <lpr>
    <max_plates value="1"/>
    <lp_filter value="1"/>
    <duplicate>
      <timeout value="10"/>
      <similarity value="100"/>
    </duplicate>
    <lp_pos_limit value="1"/>
    <min_lp_dt value="0"/>
    <max_lpcnt value="4"/>
    <anpr_frame value="0"/>
    <debug value="0"/>
    <anpr0 value="cffengines/presel/engine0"/>
    <anpr1 value="cff_auto"/>
    <def_speed_ix value="2"/>
    <min_frames value="0"/>
    <lrtext value="0"/>
    <use_trg_lpinfo value="0"/>
  </lpr>
  <proc>
    <event_timeoutms value="1000"/>
    <fifo_maxitems value="300"/>
    <event_max_frames value="16"/>
    <priority value="1"/>
  </proc>
  <proc_units>
    <quick_limit value="6"/>
    <nthreads value="2"/>
    <mtinlen value="30"/>
    <mtoutlen value="30"/>
  </proc_units>
  <syntax>
    <syntax_cnt value="0"/>
    <syntax00>
      <syntax value="LLLDDD"/>
      <syntax_weight value="1"/>
    </syntax00>
    <syntax01>
      <syntax value=""/>
      <syntax_weight value="1"/>
    </syntax01>
  </syntax>
  <syntax_grp value="presel/syntax"/>
</presel>
```

- *lpr / max\_plates* - Max. search for the license plate in the picture (FindNext () calls, 1 - no FindNext ()),
  - if we look at several license plates in one picture, the frame of the ANPR stage should be an LP frame so that you know where the 2nd, etc. should be search for a license plate, otherwise ANPR will find the one that suits the most.
  - for the same reason, reading the ADR is not possible in this case, there is a detailed explanation in the DR description
- *lpr / lp\_filter* - Filter results

<i>lp_filter</i> value	function
0	do not discard, if there is any license plate
1	No filtering
2	discard if there is no type
3	discard if only the license plate found in the preselection stage (this allows us to keep the result, for example, if ANPR Main found it, but no type)

Important: if the *use\_trg\_lpinfo* flag is set, the license plate number found by the trigger (vvq) will also be considered a license plate number, so even if set to 0, the system will not discard it, since there is no license plate text, *n.a.* will be included instead. If you don't want to keep these, set the filter of the last ANPR stage to 3, which filters such events (only vvq found something on it, not the next ANPRs), it discards.

- *lpr / duplicate / timeout* - Do not return two matching license plates within the specified time. In seconds, the filter is inactive when set to zero.
- *lpr / duplicate / similarity* - In which case we consider the license plates to be identical (in%).
- *lpr / lp\_pos\_limit* - The minimum distance of the license plate from the edge of the image in pixels, if closer, is ignored. You'll probably never need it.
- *lpr / min\_lp\_dt* - Millisecond value, if two license plates are closer than this, they will probably match. You'll probably never need it.
- *lpr / max\_lpcnt* - If we find so many same license plates, we will not examine the other images of the passage.
- *lpr / anpr\_frame* - Limit the location of the license plate search within the image (separate multiple license plates, increase reading speed). Use the full image if the frame is invalid.

<i>anpr_frame</i> value	function
0	ANPR is running in full screen
1	is the motdet frame given by the camera
2	trigger frames (radar, virtual loop, vvq triggers)
3	manually entered frames
4	License plate frame found by previous stages
5	Using a frame found by a vehicle detector (hydet)

- *lpr / anpr0* - Property path to the ANPR engine.
- *lpr / anpr1* - Property path to the associated ANPR engine. (See *proc\_units / quick\_limit*)
- *lpr / def\_speed\_ix* - Help when using radar. We maintain a table in which we record the image of the license plate, grouped by speed. If there is a pass that we could not read, we find out from this table that how many images we 'usually' return at the speeds associated with the pass, and we forward it as a selected image. This is the default value of this index.
- *lpr / min\_frames* - examine at least as many images from the passage, don't close it until it's done
- *lpr / lrtext* - reverse the writing direction of Arabic plates
- *lpr / use\_trg\_lpinfo* - Do not run your own engine, but accept the license plate text and frame from the trigger (typically running vvq on the camera).
- *proc / event\_timeoutms* - Refers to the time elapsed since the event was received, in milliseconds. If this is exceeded, the event will be closed / forwarded. This limits, how much time the stage can spend processing a passage.
- *proc / fifo\_maxitems* - Stage storage size (in image)
- *proc / event\_max\_frames* - The maximum number of images per event
- *proc / priority* - Stage line priority
- *proc\_units / quick\_limit* - If the number of queued images for ANPR exceeds this, we will switch to the faster, associated ANPR engine (*lpr / anpr1*).
- *proc\_units / nthreads* - Number of ANPR processing threads
- *proc\_units / mtinlen* - ANPR thread storage size
- *proc\_units / mtoutlen* - The size of the ANPR threads
- *syntax\_grp* - Location of syntax checking properties.
- *syntax / syntax\_cnt* - How many syntaxes have been specified
- *syntax / syntax00 / syntax* - Syntax L - letter, D - digit, pl LLLDDD is the Hungarian license plate.
- *syntax / syntax00 / syntax\_weight* - Syntax weight (how many points the license plate gets if it matches the given syntax)



## ANPR Main

type: *anpr\_main*

It is advisable to do this after the preselection stage. Runs a type engine (ANPR, ADR, etc.) on the incoming image. It is possible to bond several of them in a row, e.g. one *eur* after *hun-local*. In this case, the *eur* does not examine the passes already typed by *hun-local*, it only passes them on, but if there is no type, it runs the slower *eur* engine.

```
<recognize>
  <lpr>
    <anpr_frame value="0"/>
    <lp_filter value="1"/>
    <duplicate>
      <timeout value="10"/>
      <similarity value="100"/>
    </duplicate>
    <anpr0 value="cffengines/recognize/engine0"/>
    <adr value="0"/>
    <lrtext value="0"/>
    <min_confidence value="-1"/>
  </lpr>
  <proc_units>
    <nthreads value="2"/>
  </proc_units>
</recognize>
```

- *lpr / anpr\_frame* - See above
- *lpr / lp\_filter* - See above
- *lpr / duplicate / timeout* - See above
- *lpr / duplicate / similarity* - See above
- *lpr / anpr0* - See above
- *lpr / adr* - Enable ADR reading

<i>adr</i> value	function
0	none
1	EADR read only
2	ADR read only
3	EADR and ADR readings

The important thing is that you read the ADR in the same frame as the license plate, so if you pass on a license plate frame that has ANPR there and does not contain the ADR table, it will never find it. If you find an ADR table in an image, you are no longer looking for an EADR table on it. If you are looking for more than one license plate in an image (*presel / max\_plates*), the normal ANPR will run on the *presel* license plate, so you won't find the ADR table. In this case, there would be confusion to which plate belongs to which license plate. (Not not the one you are closer to.)

- *lpr / lrtext* - See above
- *lpr / min\_confidence* - Required ANPR confidence, discarding under it. Can be turned off with -1.
- *proc\_units / nthreads* - See above

## CMHighway

type: *cmhway*

It is able to estimate the speed optically and check the seat belt. Must be done after the ANPR Main stage because it needs (more) ANPR results. The trigger must also be set to send some pictures where you can find some license plates.

```

<cffcmhway>
  <opt_speed_enabled value="1"/>
  <check_belt_enabled value="1"/>
</cffcmhway>

```

- *opt\_speed\_enabled* - whether to estimate optical speed
- *check\_belt\_enabled* - whether a seat belt check is required

## Make and Model (MMR)

type: *mmr*

Identifies the brand and type of the vehicle. It is only worth doing after the ANPR Main stage, because it needs an ANPR result, it is worth running on the selected image (representing the event). It has no extra settings.

## Vehicle Detector

type: *hyd*

This stage can be placed in front of the presel stage as a pre-filter or as a post-filter, e.g. to filter out false positive events after ANPR. This way, we can filter out false license plates (e.g. billboards, graffiti, fences, cornfields, etc.) but do not throw away events that have a vehicle on it but no license plate, or we could not read the license plate.

```

<cffhyd>
  <filter value="0"/>
  <engine value="hydetect-7.2.0.0"/>
  <minsize value="0.100000"/>
  <maxsize value="0.900000"/>
  <minsizey value="0.100000"/>
  <maxsizey value="0.900000"/>
</cffhyd>

```

- *filter* - you can filter out events based on the result

<i>filter</i> value	function
0	no filter, it allows everything
1	if there is no vehicle and no license plate, discard the passage
2	if there is no vehicle, discard the passage
3	if there is no vehicle and no standard license plate, discard the passage

- *engine* - the location of the hydetect engine in the property tree
- *minsize*, *etc.* - the minimum / maximum x or y object size specified in % of the image size

## Postprocessing

type: *post*

Post - processing.

```

<postproc>
  <lp_img_en value="15"/>
  <titler>
    <enabled value="1"/>
    <nlines value="1"/>
    <title_string00 value="$y-$o-$d $h:$m:$s LP: $p COUNTRY: $f STATE: $j"/>
    <title_string01 value=""/>
    <title_string02 value=""/>
    <ID value="Smart/SpeedCAM"/>
    <location value="Test location"/>
    <direction_string00 value="UNKNOWN"/>
    <direction_string01 value="APPROACHING"/>
    <direction_string02 value="LEAVING"/>
    <alt_charset value="0"/>
  </titler>
  <titler_grp value="postproc/titler"/>
</postproc>

```

- *postproc / lp\_img\_en* - create a cropped license plate image
- *postproc / titler / enabled* - whether to have captions
- *postproc / titler / nlines* - how many lines (max. 8) the caption should be
- *postproc / titler / title\_string00* - caption first line
- *postproc / titler / ID* - user ID, this can be added to the caption
- *postproc / titler / location* - user location ID
- *postproc / titler / direction\_string00* - for radar, user direction identifier (unknown), this may be added to the caption
- *postproc / titler / direction\_string01* - for radar, user direction identifier (approaching), this may be added to the caption
- *postproc / titler / direction\_string02* - for radar, user direction identifier (leaving), this may be added to the caption
- *postproc / titler / alt\_charset* - caption alternative font
- *postproc / titler / titler\_grp* - subtitle property group



## 2.28. RESULT UPLOAD (ONLY SMART CAMERAS)

**Module:** lpr/cff

**Manual:** Ch. 7.2.14 Result Upload

### Note

This module will deliver its response in the binary part either as an XML document or a JPEG file.

Upload settings can be managed by setting and querying properties with the setproperty and getproperty commands [#19.Set/Get ANPR properties](#)

The properties of the upload module reside in the 'default/cfs/uplm' property path:

```
<uplm>
  <general>
    <method value="-1"/>
    <maxtryconnect value="1"/>
  </general>
  <http>
    <host value="192.168.1.250"/>
    <content value="7"/>
    <timeoutms value="2000"/>
  </http>
  <ftp>
    <host value="192.168.1.250"/>
    <content value="7"/>
    <timeoutms value="2000"/>
    <username value="user"/>
    <password value="pass"/>
  </ftp>
</uplm>
```



'general/method': Upload method.

Possible values:

- -1: disabled
- 0: HTTP/HTTPS
- 1: FTP.
- 2: SFTP

'general/maxtryconnect': Max. number of connection attempts. Possible values: 1-10

'content': Bitfield that describes what data should be uploaded. Possible values: 1-7

- bit 0: image (jpg)
- bit 1: plate image (jpg)
- bit 2: result (xml)

example:

```
all data -> content=7 (=0b111),  
image+result -> content=5 (=0b101)
```

'timeoutms': Timeout of one attempt in milliseconds.

Examples:

SET 'maxtryconnect':

```
http://IP/lpr/cff?cmd=setproperty&name=/default/cfs/uplm/general/maxtryconnect&value=5
```

GET 'maxtryconnect':

```
http://IP/lpr/cff?cmd=getproperty&name=/default/cfs/uplm/general/maxtryconnect
```

ANSWER:

```
<ans>  
<property>  
  <maxtryconnect value="1"/>  
</property>  
</ans>
```

Possible error messages in ANPR LOG:

- UploadMan:: Cannot initialize curl.
- UploadMan:: Upload is disabled.
- UploadMan:: No HTTP host defined.
- UploadMan:: No FTP host defined.
- UploadMan:: No FTP username defined.

## 2.29. CLEAR DATABASE/REMOVE RECORD (SMART CAMERAS ONLY)

**Module:** lpr/cff

**Manual:** Ch. 7.2.1 Browse

### Note

This module will deliver its response in the binary part either as an XML document or a JPEG file.

Clear database or remove record

GET:

[http://camera\\_ip/lpr/cff?cmd=cleardb](http://camera_ip/lpr/cff?cmd=cleardb)

[http://camera\\_ip/lpr/cff?cmd=removebyid&id=1396465617988](http://camera_ip/lpr/cff?cmd=removebyid&id=1396465617988)

ANSWER

```
<ans>
<ecode value="0"/>
</ans>
```

## 2.30. ANPR MODULE RESTART (SMART CAMERAS ONLY)

**Module:** lpr/cff

**Manual:** Ch. 8.7 Restart & Recovery Mode

### Note

This module will deliver its response in the binary part either as an XML document or a JPEG file.

Restart license plate reading module

GET

[http://\[camera\\_ip\]/lpr/cff?cmd=restart](http://[camera_ip]/lpr/cff?cmd=restart)

ANSWER:

```
<ans>\r\n
<ecode value="0"/>\r\n
</ans>
```

## 2.31. SYSTEM INFORMATION

**Module:** stat/identify, stat/system, hwlayer/cperiph

**Manual:** Ch. 8.1 System Info

Get information about system diagnostic values

GET:

[http://camera\\_ip/stat/identify?getall&wfilter=X](http://camera_ip/stat/identify?getall&wfilter=X)

[http://camera\\_ip/stat/system?getall&section=memory&wfilter=X](http://camera_ip/stat/system?getall&section=memory&wfilter=X)

[http://camera\\_ip/hwlayer/cperiph?getcamtemp&panel=0&wfilter=X](http://camera_ip/hwlayer/cperiph?getcamtemp&panel=0&wfilter=X)

[http://camera\\_ip/hwlayer/sensors?getEv&device=lightsensor&wfilter=X](http://camera_ip/hwlayer/sensors?getEv&device=lightsensor&wfilter=X)



## 2.32. CAMERA LOG

**Module:** logging/logstream

**Manual:** Ch. 8.3 Camera Log

### Get camera log

GET:

[http://camera\\_ip/logging/logstream?getfilter&wfilter=X](http://camera_ip/logging/logstream?getfilter&wfilter=X)

[http://camera\\_ip/logging/logstream?getlog](http://camera_ip/logging/logstream?getlog)

#### Note

- Lines are separated by '\n' (0x0a characters)
- The log will not be empty after the query, so the queried lines will remain even after the next query, unless it was not overwritten  
(It is possible to set the already queried rows to be deleted.  
See: plainconfig mode parameter)

SET:

[http://camera\\_ip/logging/logstream?filter=uarttrigger&save&wfilter=X](http://camera_ip/logging/logstream?filter=uarttrigger&save&wfilter=X)

[http://camera\\_ip/logging/logstream?filter=\\*&save&wfilter=X](http://camera_ip/logging/logstream?filter=*&save&wfilter=X)

#### Note

- '\*' no filtering is applied

### Only Smart cameras:

GET:

[http://camera\\_ip/lpr/cff?cmd=getcontent&name=/mnt/data/cfflow/cff.log](http://camera_ip/lpr/cff?cmd=getcontent&name=/mnt/data/cfflow/cff.log)

#### Note

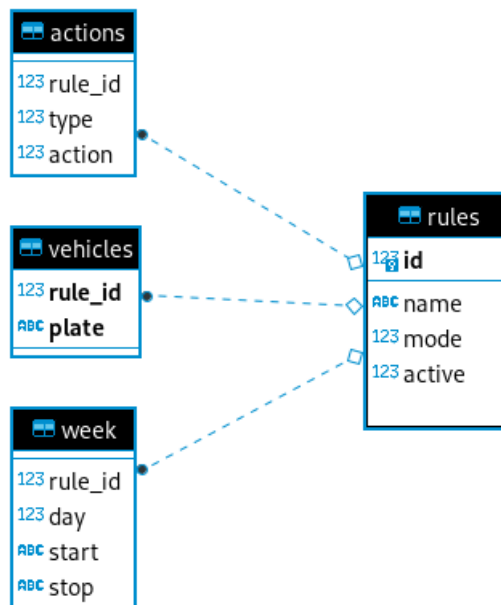
- The answer is base64 encoded.

## 2.33. BLACK/WHITELIST (SMART CAMERAS ONLY)

Black- / Whitelist functionality allows us to perform tasks depending on the time and the license plate read. It can be enabled with the `/ default / cfs / db / bw / enabled` property. Tasks can be:

- Upload (HTTP, FTP, etc.)
- GDS insertion
- run script (eg to pull GP output)
- write to websocket, TCP port
- etc.

Configurable SQL commands are used next to the table structure below.



The *rule* table:

- **id** - rule identifier, the other tables refer to it
- **name**
- **mode** - whether black (0) or white (1) list
- **active** - is the given rule active (1)

The *week* table:

- **rule\_id** - which rule the time limit belongs to
- **day** - 0-6 to which of the days of the week it applies (0-Sunday, 1-Monday, ... 6-Saturday)
- **start** - between 0:00 and 24:00 when the active time interval starts
- **end** - between 0:00 and 24:00 when the active time interval ends

The *vehicles* sign:

- **rule\_id** - which rule the license plate belongs to
- **plate** - license plate text

The *actions* table:

- **rule\_id** - which rule the action belongs to
- **type** - what type of event, possible task types:

Type	Name
0	Upload (Upload Manager)
1	Execute script
2	Stream
3	Datadisplay (websocket)
4	GDS Insertion

- **action** - within which (eg which from several UploadManagers - for example, in case of 3, it uses the Upload Manager configured under `* default / cfs / uplm / extras / uplm3`)

## Principle of operation

Read a **license plate** at a **given time**. Look all the rules in the rules table to see if any of them apply to a given time (day and time are correct). If yes, check the vehicles license plate to see if it has the license plate text. After that, depending on whether the rule was recorded in black or whitelist mode, we execute (black) or do not execute (white), the task added to the rule in the action table. If no such rule is found and there is a rule called default, the corresponding action is executed independently of the license plate text.

### Note

The mode parameter is only present in newer releases for compatibility, it does nothing, so it can only be used in mode 0. Whitelist functionality can be implemented with rules where we do not assign a task to the listed license plates (whitelist, we do not penalize them, eg ambulance on the highway), and we define a default rule for all others (they can be penalized).

## Commands

Tables can be written with SQL commands (cmd = querydb) using the following examples:

- New rule

```
INSERT INTO rules (name) VALUES("[rule name"])
```

- Activate a rule

```
UPDATE rules SET active="1" WHERE id="[rule ID"])
```

- Add a task to a rule

```
INSERT INTO actions (rule_id,type,action) VALUES ("[rule id]", "[type]", "[action index]")
```

- Add a date to a rule

```
INSERT INTO week VALUES("[rule id"],["weekday"],["start date"],["end date"])
```

- Add a vehicle to a rule

```
INSERT INTO vehicles VALUES("[rule id"],["license plate text"])
```

## Bulk filling

There is a cmd = uploadtable command that can be used to upload an entire table separated by '\n', e.g. if you want to add many license plates to a rule. This is how it should look:

```
[table name]  
[key (column names listed in parentheses)]  
[values 0 (column values separated by commas)]  
[values 1 (comma-separated column values)]  
(...)  
[values N (column values separated by commas)]
```

For example:

```
vehicles  
(rule_id,plate)  
"2","ABC123"  
"2","DEF456"  
"2","GHI678"  
"2","JKL901"  
"2","MNO234"
```



## 2.34. RESTART

Restart the camera

GET:

[http://camera\\_ip/setup/exit?retcode=1](http://camera_ip/setup/exit?retcode=1)

### Note

Return codes (retcode) available:

- 1: restart modules (quick restart)
- 3: reboot camera
- 7: reboot and enter recovery mode

## 2.35. CAMERA MODULES

List all modules currently loaded

GET:

[http://camera\\_ip/stat/system?getall&section=module&wfilter=X](http://camera_ip/stat/system?getall&section=module&wfilter=X)

ANSWER:

```
...
mn00=logging/logstream\r\n
mv00=3.1.0.6,0\r\n
md00=Device Log Streamer\r\n
mn01=maintenance/backup\r\n
mv01=3.1.1.4,2\r\n
md01=Backup System\r\n
...
```

General syntax:

```
mnXX=group of the module/ name of module\r\n
mvXX=version of the module\r\n
mdXX=description of the module\r\n
where XX is the index of the module
```

## Contact Information

### Headquarters:

Adaptive Recognition Hungary Inc.  
Alkotás utca 41 HU-  
1123 Budapest Hungary  
Phone: +36 1 201 9650  
Fax: +36 1 201 9651

Web: [www.adaptiverecognition.com](http://www.adaptiverecognition.com)

### Service Address:

Adaptive Recognition Hungary Inc.  
Ipari Park HRSZ1113/1 HU  
2074 Perbál Hungary  
Phone: +36 1 2019650

E-mail: [rmarequest@adaptiverecognition.com](mailto:rmarequest@adaptiverecognition.com)

Adaptive Recognition Hungary Technical Support System (ATSS) is designed to provide you the fastest and most proficient assistance, so you can quickly get back to business. For further technical information about our products, please visit our official website.

Information regarding hardware, software, manuals and FAQ are easily accessible for customers who previously registered to enter the dedicated ATSS site. Besides offering assistance, the site is also designed to provide maximum protection while managing your business information and technical solutions utilized.

### New User

If this is your first online support request, please create an account by clicking on this [link](#).

### Returning User

All registered ATSS customers receive a personal access link via e-mail. If you previously received a confirmation message from ATSS, it contains the embedded link that allows you to securely enter the support site.

If you need assistance with login or registration, please contact [atsshelf@adaptiverecognition.com](mailto:atsshelf@adaptiverecognition.com) for help.

