

TraFFic CaMMRa Application

Integration API v 1.0

Updated: 2019-10-03

Latest version always at:

Real-time events API VAPIX or ONVIF protocol	2
Event data packet structure	2
Event data packet example	2
HTTP management API	3
Events request	3
Request URL examples	3
Event response	3
Event response data example	4
HTML response	5
Command for black/white lists upload	5
API FF Integration Events	6
Http multipart post	6
TCP	6
JSON Fields	6
JSON example	8
Simple php receiver for multipart post	9
Simple Web form for testing receiver	9
Utilities	10
php-listening	10
Technical Support	10

Real-time events API VAPIX or ONVIF protocol

Event data packet structure

Real-time recognition events can be received from camera using VAPIX or ONVIF protocol.
All recognition-related data without the image is present. Packet structure is as following:

Field name	Description
text	Plate number as text
country	Alpha3 ISO3166 country code https://en.wikipedia.org/wiki/ISO_31661_alpha3
height	Plate crop height, pixels
width	Plate crop width, pixels
left	Plate crop left edge coordinate, pixels
top	Plate crop top edge coordinate, pixels
listName	“Black list” or “White list” if license plate is listed; empty otherwise
action	“Opened” if barrier was opened following app configuration; “No action” otherwise
consumedTime	License plate recognition duration, ms

Event data packet example

```

<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
<tt:Event xmlns:tt="http://www.onvif.org/ver10/schema">
<wsnt:NotificationMessage xmlns:tns1="http://www.onvif.org/ver10/topics"
xmlns:tnsaxis="http://www.axis.com/2009/event/topics" xmlns:wsnt="http://docs.oasisopen.org/wsn/b2"
xmlns:wsa5="http://www.w3.org/2005/08/addressing">
<wsnt:Topic
Dialect="http://docs.oasisopen.org/wsn/t1/TopicExpression/Simple">tnsaxis:CameraApplicationPlatform/LPR<
/wsnt:Topic>
<wsnt:ProducerReference>
<wsa5:Address>uri://94f6d41addf84a33b706e0b5733c7b9b/ProducerReference</wsa5:Address>
</wsnt:ProducerReference>
<wsnt:Message>
<tt:Message UtcTime="20160519T08:25:07.988906Z">
<tt:Source>
</tt:Source>
<tt>Data>
<tt:SimpleItem Name="height" Value="28" />
<tt:SimpleItem Name="width" Value="148" />
<tt:SimpleItem Name="country" Value="FRA" />
<tt:SimpleItem Name="left" Value="701" />
<tt:SimpleItem Name="top" Value="581" />
<tt:SimpleItem Name="consumedTime" Value="801" />

```

```
<tt:SimpleItem Name="action" Value="" />
<tt:SimpleItem Name="listName" Value="" />
<tt:SimpleItem Name="text" Value="996XA49" />
</tt:Data>
</tt:Message>
</wsnt:Message>
</wsnt:NotificationMessage>
</tt:Event>
</tt:MetadataStream>
```

HTTP management API

Events request

Download event records from on-camera database.

Any request parameter can be employed independently of others, string look-ups are case insensitive. Fields of `string` type can be matched against partial sequence from request parameter.

HTTP GET request target:

<http://<camera-IP-address>/local/fflprapp/search.cgi?<paramName1>=<value1>&<paramName2>=<value2>>

Parameter	Description
text	Number plate text or a part thereof (e.g. AA11 for plates AA1122BB or AA1134XX)
limit	Limit of events to return (like SQL LIMIT clause)
offset	Records to skip before results get returned (like SQL OFFSET clause)
timestampFrom	Timestamp to select events after the given value. Linux timestamp, milliseconds. May be used without "timestampTo".
timestampTo	Timestamp to select events happened before the given value. Linux timestamp, UTC, milliseconds. May be used without "timestampFrom".
country	Alpha3 ISO3166 country code, or a part thereof
action	0 (return only "no action" events) or 1 (return only "opened" events); skip to receive all events

Request URL examples

<http://10.0.3.100/local/fflprapp/search.cgi?timestampfrom=1467648840000000×tampto=1467648874652000&text=&action=&country=&limit=11&offset=0>

<http://10.0.3.100/local/fflprapp/search.cgi?limit=20>

Event response

Response contains XML document, data fields are as follows:

Field	Description
TS	Event timestamp, Unix timestamp, UTC, milliseconds
MOD_TS	Event timestamp, human readable
CAR_ID	Vehicle ID, per database id
LPR	Number plate as text
RTIME	License plate recognition duration, ms

ACTION	Barrier action taken : 0= "no action", 1 = "opened"
ACT_PARAM	<i>deprecated</i>
THRESHOLD	Recognition confidence value, float
ROI_X	ROI coordinates and dimensions
ROI_Y	
ROI_W	
ROI_H	
LP_X	Number plate crop coordinates and dimensions
LP_Y	
LP_W	
LP_H	
LP_BMP	URL of a plate image stored on a camera
ROI_BMP	URL of a ROI image stored on a camera, if enabled
COUNTRY	Alpha3 ISO3166 country code https://en.wikipedia.org/wiki/ISO_31661_alpha3
ROI_ID	ROI number, ACAP is supporting 2 ROI in the frame
ROI_IDU	reserved
FRAMES	reserved
DISTANCE	reserved
SPEED	reserved
DIRECTION	2 IN 3 OUT 0,1- UNKNOWN

Event response data example

```
<event>
```

```

<TS>1528371395804689</TS>
<MOD_TS>2018-06-07 14:36:36 TRT</MOD_TS>
<CAR_ID>66948</CAR_ID>
<LPR>2AN2563</LPR>
<RTIME>150.0</RTIME>
<ACTION>0</ACTION>
<ACT_PARAM></ACT_PARAM>
<THRESHOLD>0.717648804187775</THRESHOLD>
<ROI_X>53</ROI_X>
<ROI_Y>54</ROI_Y>
<ROI_W>1728</ROI_W>
<ROI_H>972</ROI_H>
<LP_X>1501</LP_X>
<LP_Y>288</LP_Y>
<LP_W>168</LP_W>
<LP_H>28</LP_H>
<ROI_ID>1</ROI_ID>
<ROI_IDU>0</ROI_IDU>
<FRAMES>0</FRAMES>
<DISTANCE>0</DISTANCE>

```

```
<SPEED>0</SPEED>
<DIRECTION>1</DIRECTION>
<LP_BMP>tools.cgi?action=getImage&name=45/20180607143636_205385lp_2AN2563_2176954.png</LP_BMP>
<ROI_BMP>tools.cgi?action=getImage&name=44/20180607143635_805012lp_2AN2563_2176951.png</ROI_BMP>
<COUNTRY>CZE</COUNTRY>
</event>
```

HTML response

HTTP GET request target:

`http://<camera-IP-address>/local/fflprapp/events.cgi?<paramName1>=<value1>&<paramName2>=<value2>`

Request and response parameters are as above, while response data is wrapped into HTML instead of XML.

Command for black/white lists upload

```
curl -v --anyauth "http://root:pass1234@10.0.3.173/local/fflprapp/white_list.cgi" -s -d@white.xml
```

```
curl -v --anyauth "http://root:pass1234@10.0.3.173/local/fflprapp/black_list.cgi" -s -d@black.xml
```

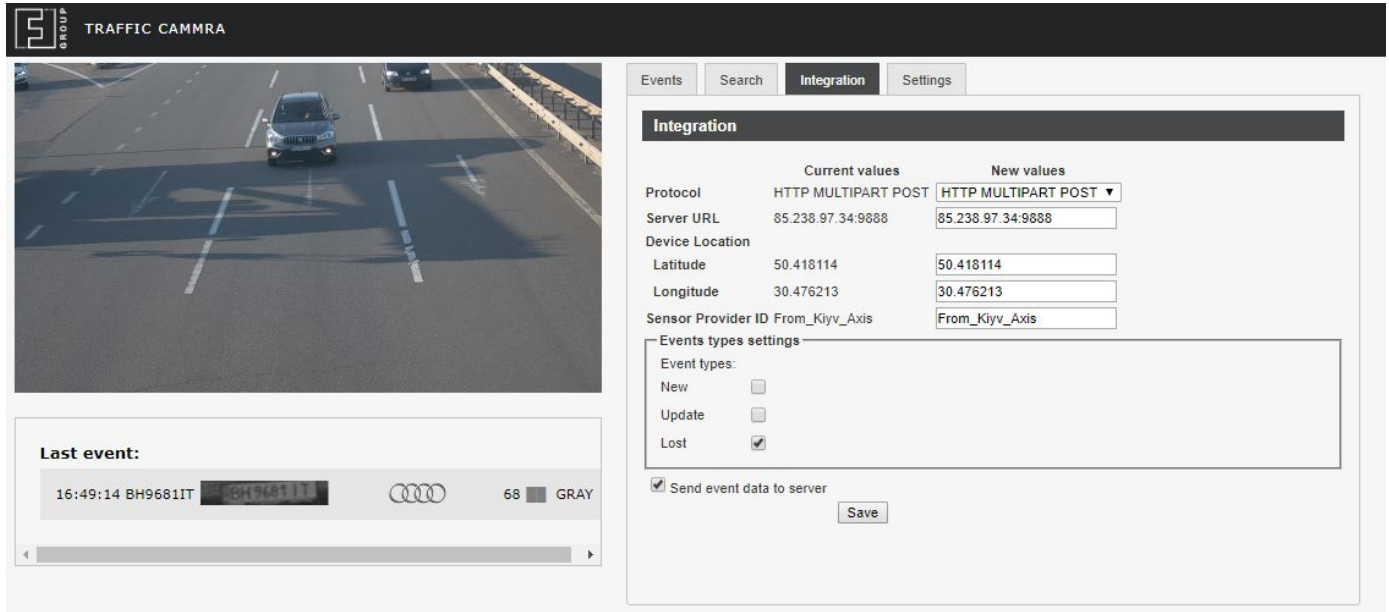
Lists examples: [black.xml](#), [white.xml](#)

Please use xml validator www.xmlvalidation.com to check your files.

Video illustration: watch [online](#), or download [AVI](#) or [MP4](#).

Windows CURL library: <https://winampplugins.co.uk/curl/>

API FF Integration Events



The screenshot displays the TRAFFIC CaMMRa web interface. On the left, a live camera feed shows a car on a road. Below the feed, a 'Last event:' section shows a timestamp '16:49:14', a license plate 'BH9681IT', a car icon, and the text '68 GRAY'. On the right, the 'Integration' settings panel is open, showing 'Current values' and 'New values' for various parameters.

	Current values	New values
Protocol	HTTP MULTIPART POST	HTTP MULTIPART POST ▼
Server URL	85.238.97.34:9888	85.238.97.34:9888
Device Location		
Latitude	50.418114	50.418114
Longitude	30.476213	30.476213
Sensor Provider ID	From_Kiyv_Axis	From_Kiyv_Axis

Below the table, the 'Events types settings' section shows checkboxes for 'New', 'Update', and 'Lost'. The 'Lost' checkbox is checked. A 'Send event data to server' checkbox is also checked. A 'Save' button is at the bottom right of the settings panel.

- **New** - This event occurs when the vehicle has been detected for the first time. The direction is undefined
- **Update** - This event occurs when the data in the package was changed and updated
 - The recognized license plate was an update of a previous one
 - The direction was calculated and added
 - Recognition zone was changed
- **Lost** - This event occurs when detected vehicle has been absent in the recognition zone more than 5s

Http multipart post

For sending JSON data to WEB service used multipart POST HTTP(s) requests.

Transferred data:

- JSON with event description
- Image

TCP

For sending data to external applications using TCP protocol through the entered **URL**. External application should open TCP socket and listen the data

Transferred data(ONLY TEXT):

- JSON with event description

JSON Fields

Token	Type	Format / Value	Description
packetCounter	Unsigned int		Counter of packets. Starts from "1". Counter starts after successful connection or after start of device. It does not matter.
datetime	string	YYYYMMDD HHMMSSFFF YYYY - year 4 digits MM - month 2 digits (with leading 0) DD - day 2 digits (with leading 0)(space) HH - hours MM - minutes SS - seconds FFF - milliseconds TimeZone in form +-HHMM optional.	%Y%m%d %H%M%S%3d
plateUTF8	UTF string	"RZ3832H"	
plateUnicode	Unicode string	"RZ3832H"	
plateText	Unicode string	\u0041\u0041....	Plate number in the unicode translation https://www.branah.com/unicode-converter
plateCountry	ASCII string	[DNK,.....UKR]	Country, ISO3166, ALPHA3
plateConfidence	float		Recognition confidence
carState	ASCII string	[new,updated,lost]	Event type.
geotag: { "lat":,"lon": }	Json object with floats	50.418114,30.476213	Latitude, longitude
imageType	ASCII string	"plate" "mmrcrop" "vehicle" "frame"	The kind of attached image: plate only rectangle with plate number, frame - full frame.

plateImageType	ASCII string	[png,jpeg,bmp]	The type of attached image. HTTP header must use the same type with header Content-type: image/...
plateImageSize	Unsigned long	0	The size in bytes of attached image, may be zero.
carMoveDirection	string	[unk,in,out]	The kind of detected movement
timeProcessing	Unsigned int		Time of image processing. ms
plateCoordinates	Array of unsigned int	[left,top,width,height]	Plate coordinates
carID	Unsigned long long		Internal runtime plate ID
sensorProviderID	ASCII string		The Human name of sensor
roiID	Unsigned int	1,2	Number of region of interest. ACAP support max 2
frame_timestamp	unsigned long long	155131902113266	The returned value is nanoseconds for the system uptime(Frame time)
capture_timestamp	unsigned long long	1558090545839	Obtain the timestamp of the media_frame, measured in nanosecond. in the millisecond
capture_ts			To be deprecated
"vehicle_info":{"brand":"","model":"","type":"","color":"","confidence":""}	string	"vehicle_info":{"brand":"Ford","model":"focus","type":"CAR","color":"BLACK","confidence":"45.053627"}	Vehicle recognition date Brand Model Type Color Confidence level
imagesURI	string	["/local/fflprapp/tools.cgi?action=getImage&name=2/20191010165230_615401lp_H634TK750_210.jpg","/local/fflprapp/tools.cgi?action	List of available event images' URIs

		=getImage&name=3/20191010165230_618056roi_H634TK750_210.jpg"],	
--	--	--	--

JSON example

```
{
  "packetCounter": "210",
  "capture_timestamp": "1570715550300",
  "frame_timestamp": "1825024378796229",
  "capture_ts": "1570715550300000000",
  "current_ts": "1825024705524058",
  "current_ts2": "1570715550627640401",
  "datetime": "20191010 165230000",
  "plateText": "\u0048\u0036\u0033\u0034\u0054\u004b\u0037\u0035\u0030",
  "plateUnicode": "\u0048\u0036\u0033\u0034\u0054\u004b\u0037\u0035\u0030",
  "plateUTF8": "H634TK750",
  "plateASCII": "H634TK750",
  "plateCountry": "RUS",
  "plateConfidence": "0.731480",
  "carState": "new",
  "roiID": "2",
  "geotag": {
    "lat": 50.418114,
    "lon": 30.476213
  },
  "imageType": "plate",
  "plateImageType": "png",
  "plateImageSize": "0",
  "carMoveDirection": "unknown",
  "timeProcessing": "0",
  "plateCoordinates": [750, 487, 140, 28],
  "carID": "69298",
  "GEOtarget": "Camera",
  "imagesURI": [
    "/local/fflprapp/tools.cgi?action=getImage&name=2/20191010165230_615401lp_H634TK750_210.jpg",
    "/local/fflprapp/tools.cgi?action=getImage&name=3/20191010165230_618056roi_H634TK750_210.jpg"
  ],
  "vehicle_info": {
    "brand": "Ford",
    "model": "focus",
    "type": "CAR",
    "color": "BLACK",
    "confidence": "45.053627"
  },
  "sensorProviderID": "EventTest2"
}
```

Simple php receiver for multipart post

This simple receiver may be used with any HTTP server which support PHP extension.

ff_post.php

```
<?php
$uploadaddir = 'log/';
echo "<pre>\n";
$uploadfile = $uploadaddir . basename($_FILES['event']['name']);
if (move_uploaded_file($_FILES['event']['tmp_name'], $uploadfile)) {
    echo "File json uploaded.\n";
} else {
    echo "Upload report error!\n";
}
$uploadfile = $uploadaddir . basename($_FILES['image']['name']);
if (move_uploaded_file($_FILES['image']['tmp_name'], $uploadfile)) {
    echo "File image uploaded.\n";
} else {
    echo "Upload image error!\n";
}
echo 'Debug info:\n';
print_r($_FILES);
echo "</pre>\n";
?>
```

Simple Web form for testing receiver

ff_post_form.html

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
</head>
<form method="post" enctype="multipart/form-data" action="ff_post.php">
    Enter event file: <input type="file" name="event" size="40">
    Enter image file: <input type="file" name="image" size="40">
    <input type="submit" value="send" name="submit">
</form>
</html>
```

Utilities

php-listener

Test integrations over HTTP using local standalone WAMP web-server.

https://drive.google.com/open?id=1P-s1_J1mh-mSaRqn_A-Y5eD4mM5CgqXI:

Online docs: <https://github.com/OleksiyRudenko/php-listener/blob/master/README.md>

Technical Support

Please, address your further inquiries to acap.support@team.ff-group.org