# AI Guides

## 1 Mirasys List Management (LM)

### 1.1 Mirasys List Management Introduction

List Management (LM) is used to process Face Recognition (FR) and License Plate Recognition (LPR) events by matching detected faces and license plates to an identity and identity list. LM service is used to store identities and identity list information, receive and save LPR and FR events, send LPR and FR events to clients, do searches in saved events, and send LPR and FR events to the VMS server for processing.

LM service has the following abilities:

- Store identities and identity lists in the database
- Receive and store LPR and FR events in the database
- Match detected license plates and faces to defined identities and identity lists
- Search LPR and FR events from the database using search parameters
- Send real-time LPR and FR events for clients and recorders
- Send LPR and FR events to the VMS server for processing
- Notify clients and recorders about changes in identities and identity lists
- Enable integration to License Plate and Face recognition

The list Management service has a separate installer, so it can execute on a separate server or on some VMS server.

List Management settings (identities and identity lists) can be managed in System Manager List Management settings and in the Spotter Smart List Management plugin.

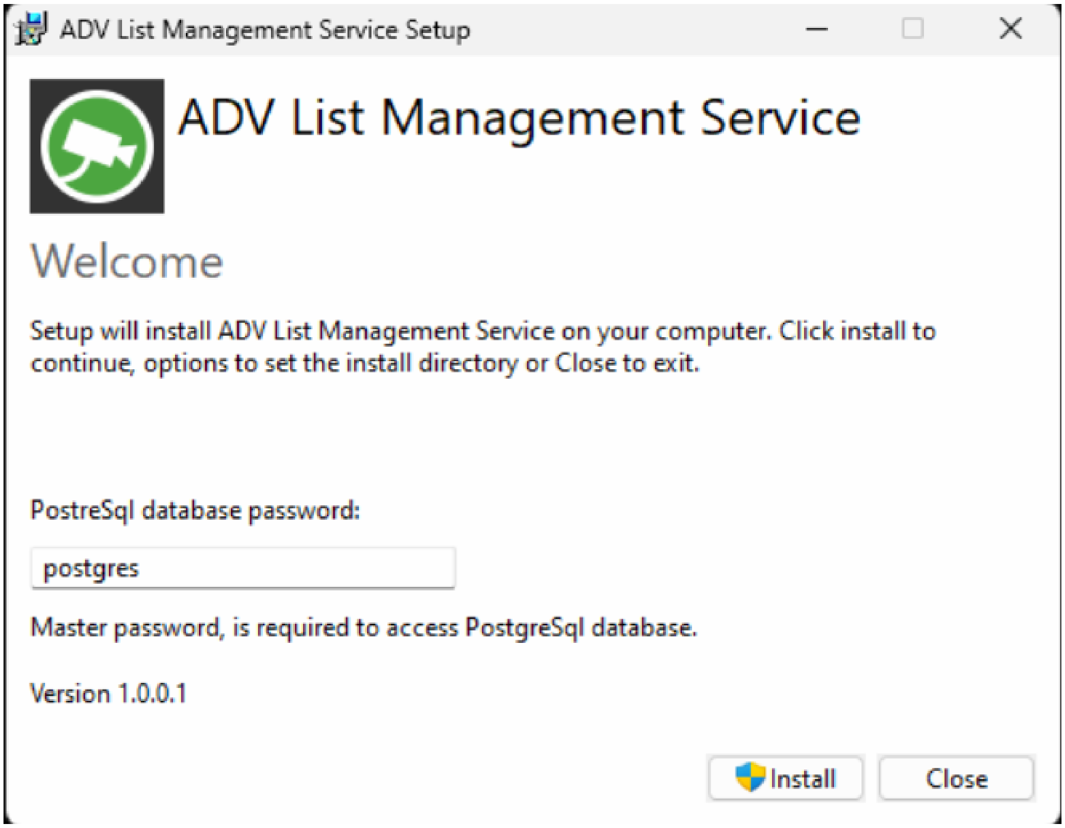
## 1.2 LM Service Installation

### 1.2.1 Requirements

- Administrator rights
- List Management Service feature is included on V9.6.0.

### 1.2.2 Installation

1. Download latest version from Extranet.
2. Unzip this example to C:\temp folder.
3. Start installation double clicking installation file.
4. Click Install to continue.
   a. Change PostgreSQL database password



ADV List Management Service Setup

ADV List Management Service

Welcome

Setup will install ADV List Management Service on your computer. Click install to continue, options to set the install directory or Close to exit.

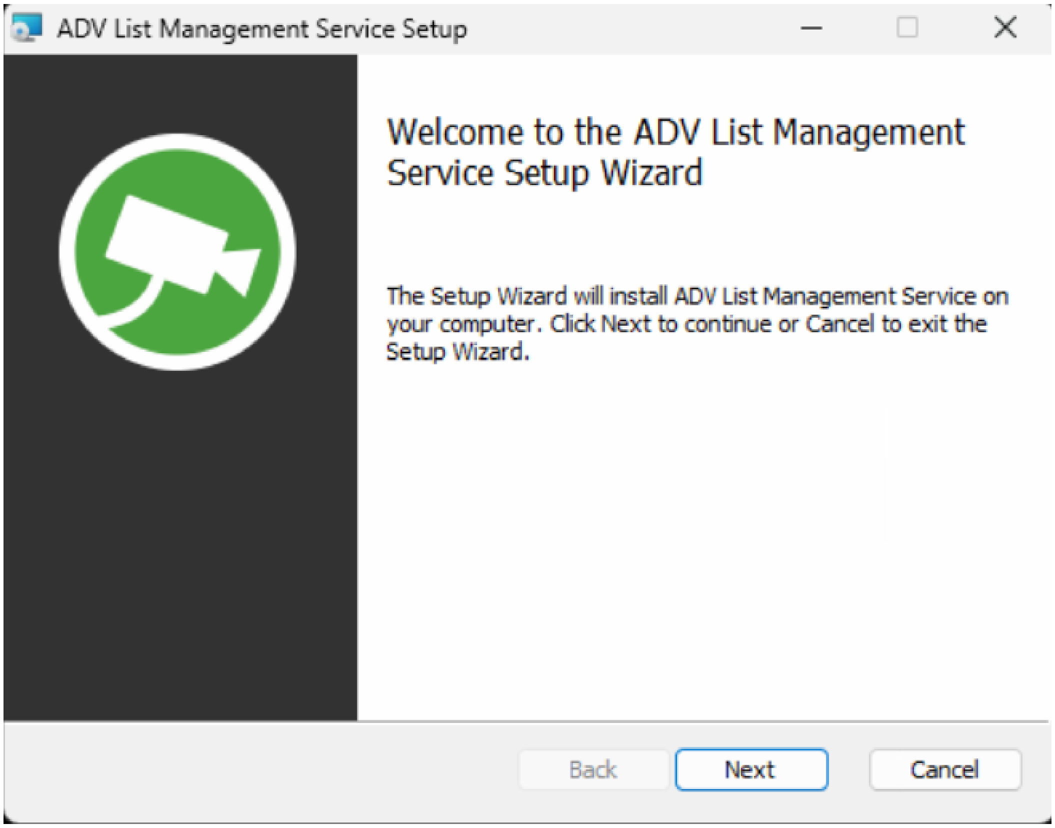
PostreSql database password:

postgres

Master password, is required to access PostgreSql database.

Version 1.0.0.1

5. Wait that PostgreSQL is installed.
    a. You made need to apply firewall rules when installation is going forward.
6. Click Next to continue.

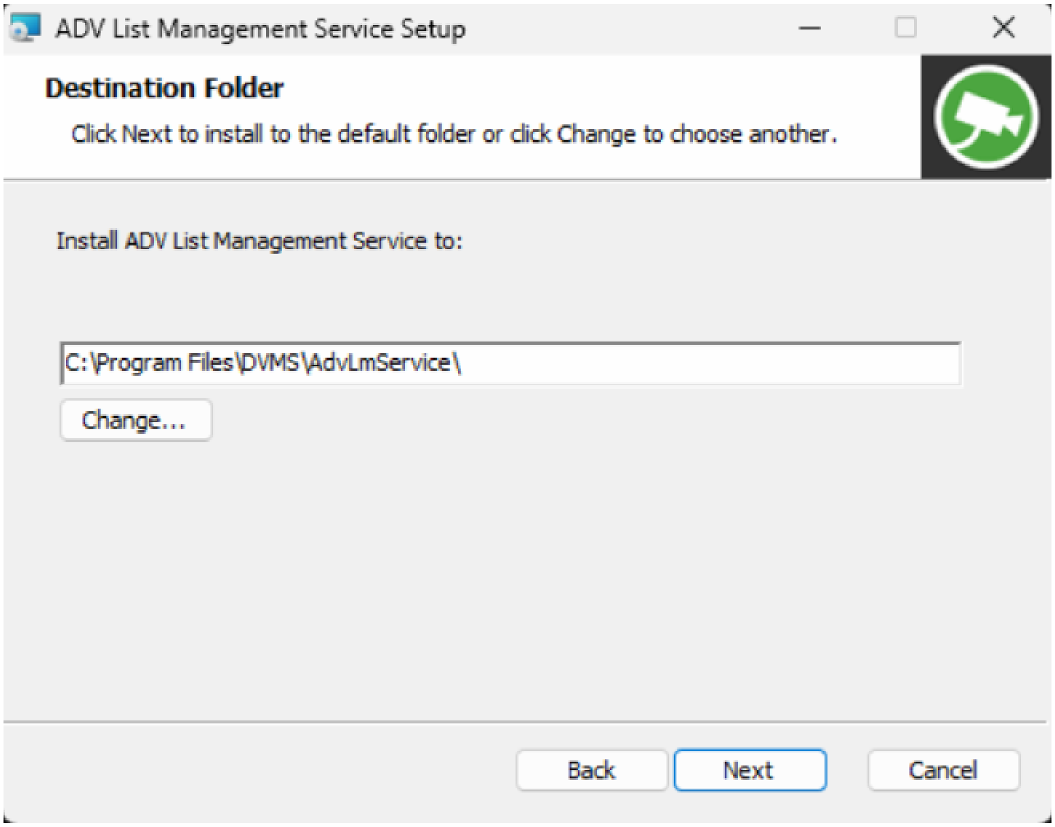


ADV List Management Service Setup

Welcome to the ADV List Management Service Setup Wizard

The Setup Wizard will install ADV List Management Service on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

7. Change installation location if needed, if not then click Next to continue.



ADV List Management Service Setup

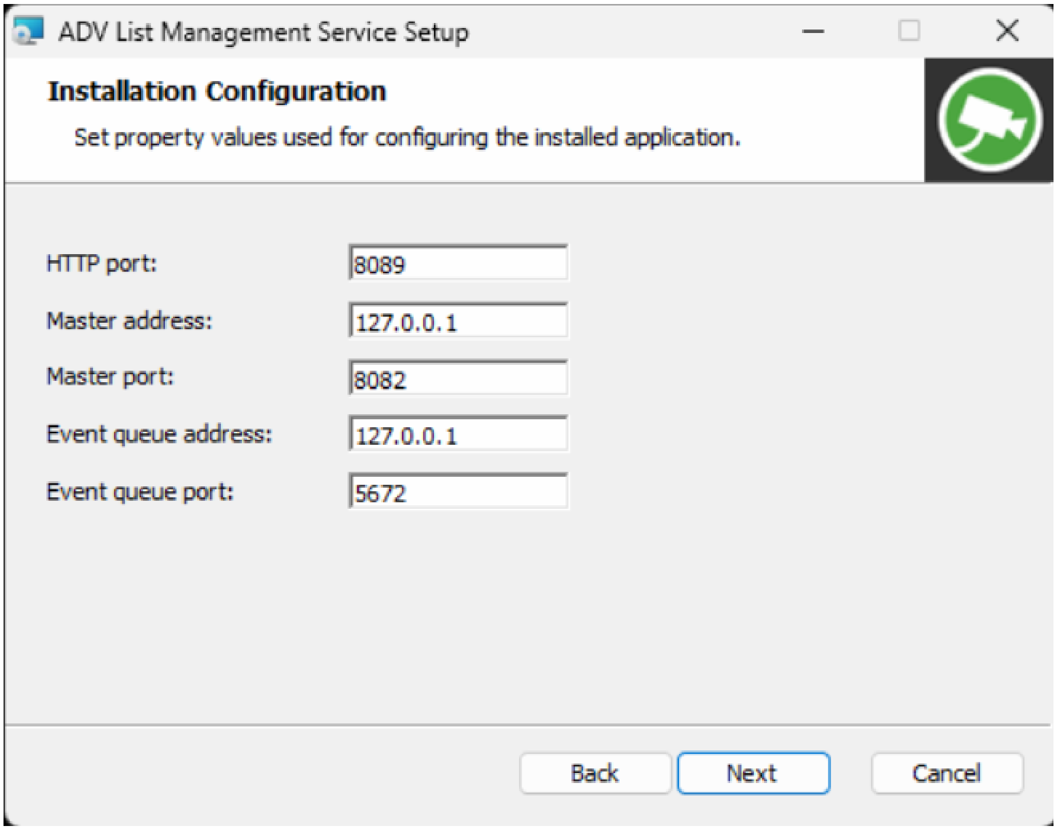
**Destination Folder**

Click Next to install to the default folder or click Change to choose another.

Install ADV List Management Service to:

C:\Program Files\DVMS\AdvLmService\

8. Change ports and addresses if needed.
    a. If you are installing List Management Service on other server, then you need to change this.
    b. Event queue address is same address where List Management Service is installed. Keep this as default.
9. Click Next to continue.



**Installation Configuration**

Set property values used for configuring the installed application.

HTTP port: 8089

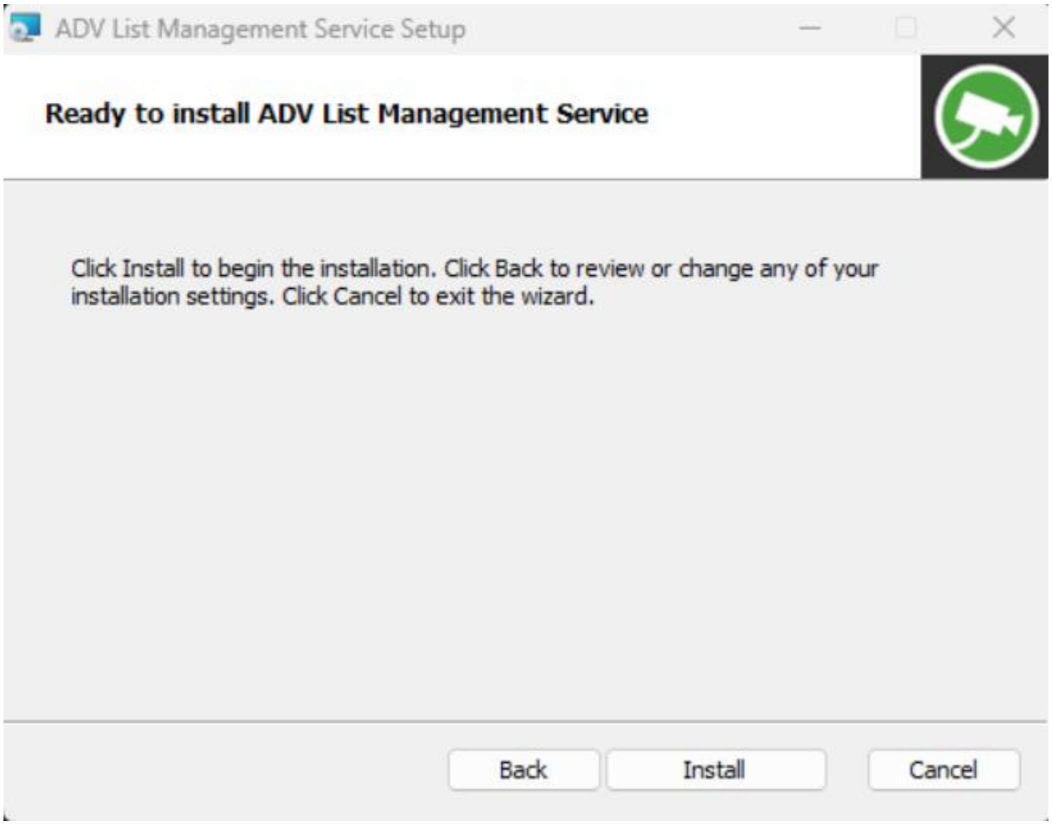
Master address: 127.0.0.1

Master port: 8082

Event queue address: 127.0.0.1

Event queue port: 5672

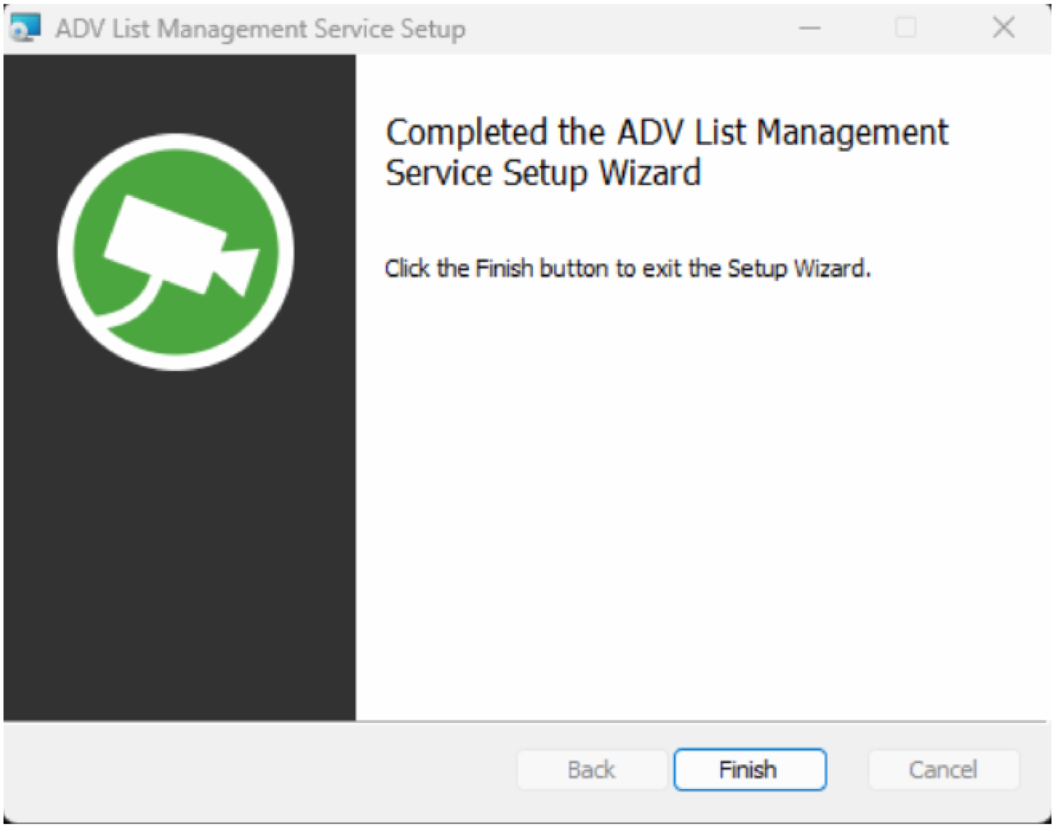
10. Click Install to continue.



Ready to install ADV List Management Service

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

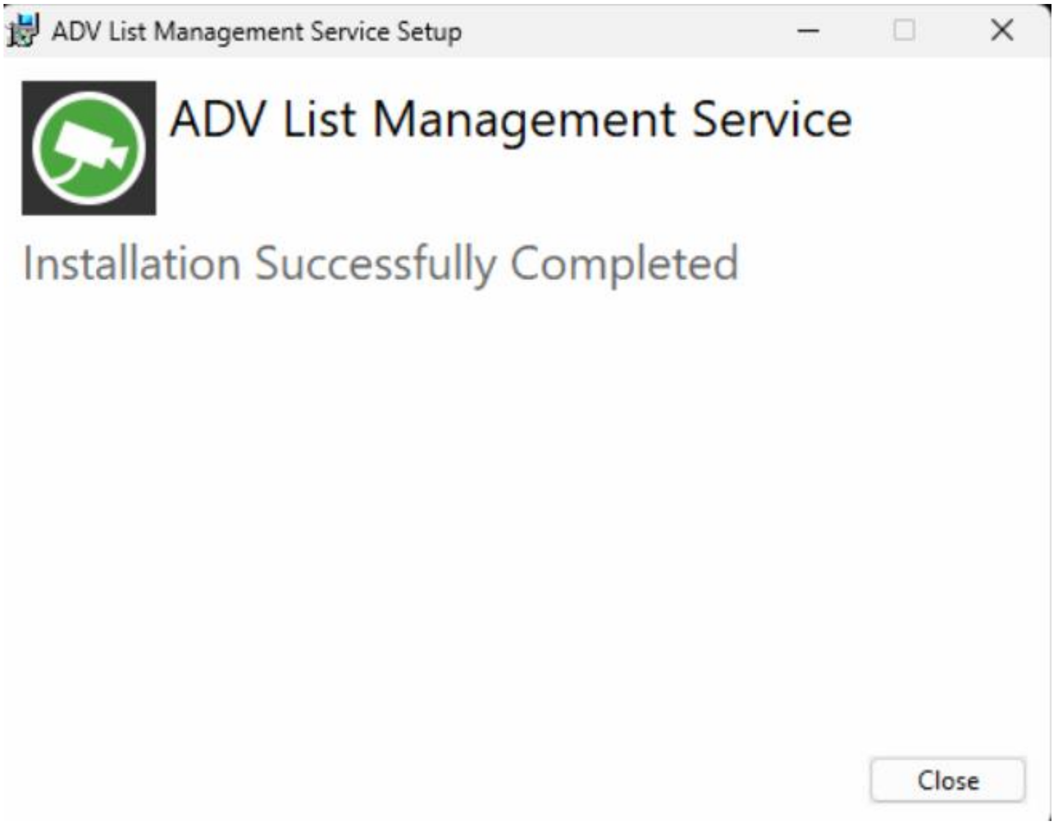
11. Wait that installation is finished.
    a. You made need to apply firewall rules when installation is going forward.
    b. Installer install RabbitMQ Server which handle events from List Management Service, Face Recognition Service and License Plate Recognition Service.
    c. Default port 5672 TCP.
12. Click Finish to end installation.



Completed the ADV List Management Service Setup Wizard

Click the Finish button to exit the Setup Wizard.

13. Click Close to close installation.



ADV List Management Service

Installation Successfully Completed

14. Now List Management Service is installed to server and ready to use.
    a. List Management Service send details to VMS Master server and you can configure service via System Manager.

# 2 Mirasys Face Recognition (FR)

## 2.1 Face Recognition Introduction

Face Recognition (FR) is used to identify a human face. It is used in the VMS System to get events when faces are detected from selected video streams and to detect when specific persons are seen in the video. Together with the Mirasys List Management, this allows you to, for example, create an automatic detection system of a person's access to the premises.

**Note that anti-spoofing is not included in version 9.6.**

The FR service receives video streams, processes images, detects faces, and sends notifications with detection data to List Management (LM) service for identity and list matching.

Face Recognition service has a separate installer, so it can execute on a separate server or on some VMS server.

Face recognition works with 112 x 112 image size. If the face is larger in the picture, it is first reduced to 112 x 112 assembled before identification is made. Likewise, if the face is smaller in the picture, it is first enlarged to that 112 x112 to size. The recommended size of the face in the image is at least 112 pixels.

On licensing side there is need to add RTSP Streaming Server feature to Management Server and all those servers where is plan to use SmartFR feature. On Management Server side there is need to be add wanted amount of channels of SmartFR feature.

## 2.2 FR Service Installation

Smart services can be used together with the VCA Deep Learning feature. In this case, you should note that you are using the latest NVIDIA drivers, and not the ones that come with the CUDA Toolkit package. More information can be found here.

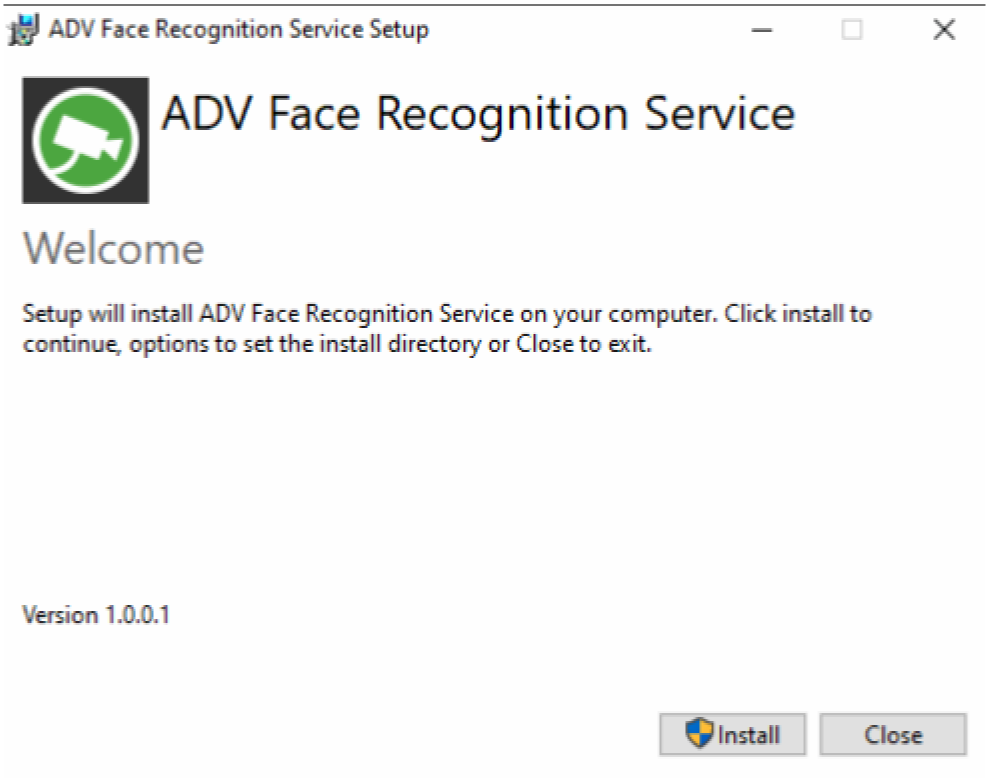
### 2.2.1 Requirements

- Administrator rights
- List Management Service installed
- Face Recognition license on VMS server

### 2.2.2 Installation

1. Download latest version from Extranet.
2. Unzip this example to C:\temp folder.
3. Start installation double clicking installation file.

4. Click Install to continue.

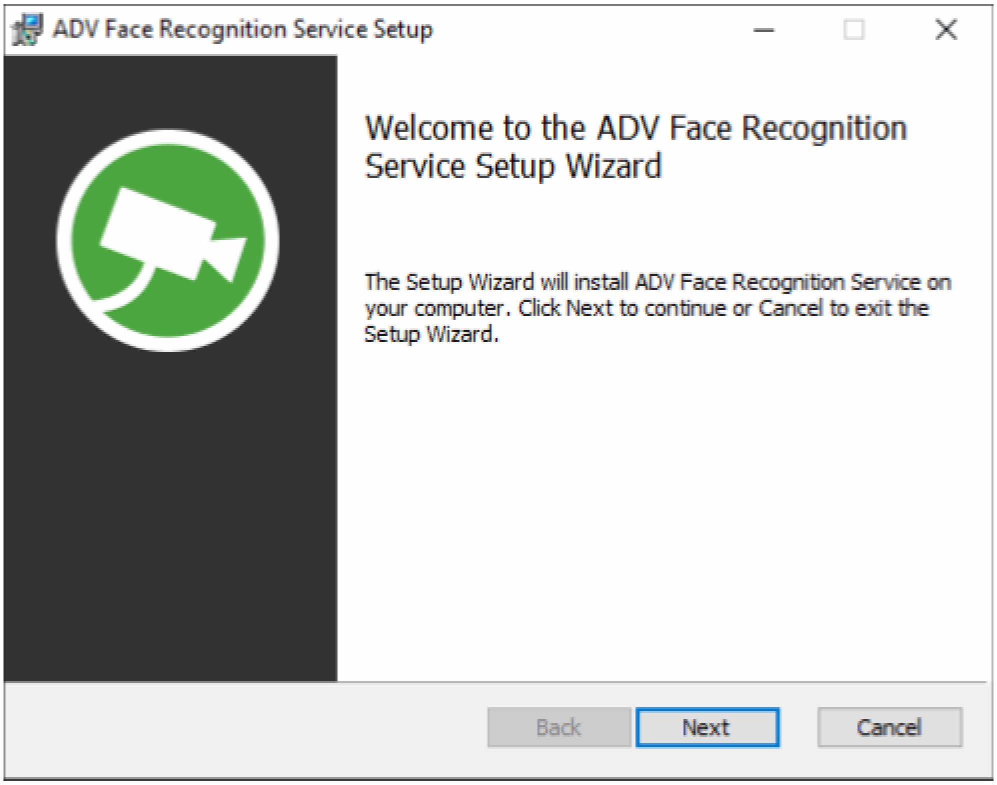


ADV Face Recognition Service Setup
ADV Face Recognition Service
Welcome

Setup will install ADV Face Recognition Service on your computer. Click install to continue, options to set the install directory or Close to exit.

Version 1.0.0.1

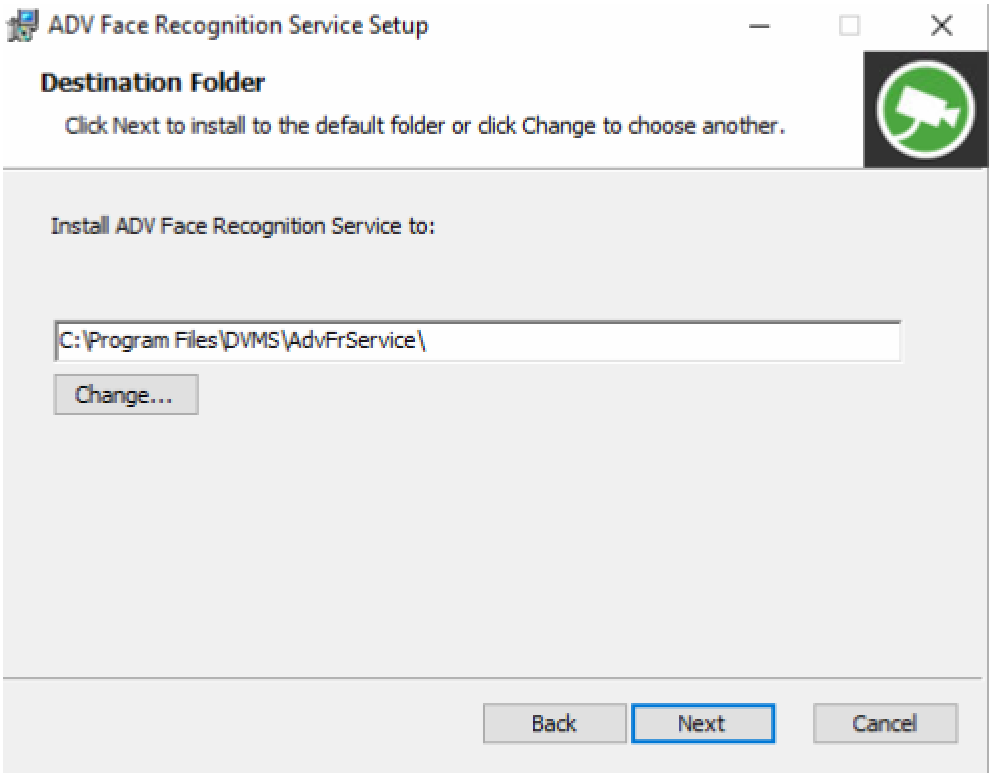
5. Click Next to continue.



Welcome to the ADV Face Recognition Service Setup Wizard

The Setup Wizard will install ADV Face Recognition Service on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

6. Change installation location if needed, if not then click Next to continue.



ADV Face Recognition Service Setup

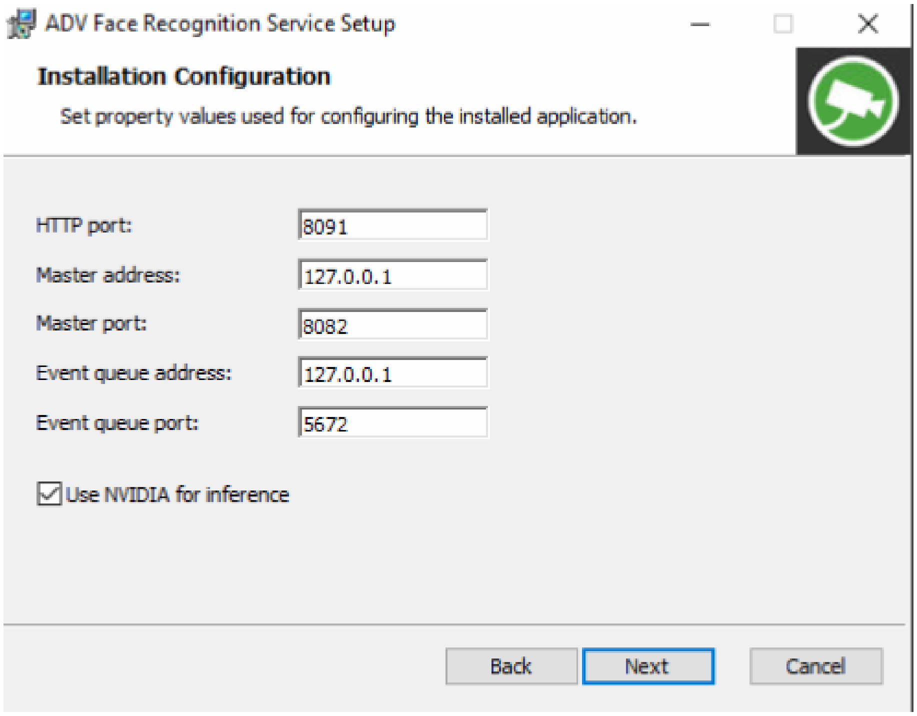
**Destination Folder**

Click Next to install to the default folder or click Change to choose another.

Install ADV Face Recognition Service to:

C:\Program Files\DVMS\AdvFrService\

7. Change ports and addresses if needed.
    a. If you example install Face Recognition service to other machine than VMS Master, you need change Master address to correct one.
    b. Same apply for Event queue address. Replace this address with that server address where List Management Service is installed.
    c. If you have Nvidia graphics card installed to server, you can keep Use NVIDIA for inference enabled. This create Nvidia models to use graphics card.
8. Click Next to continue.



**Installation Configuration**

Set property values used for configuring the installed application.

HTTP port: 8091

Master address: 127.0.0.1

Master port: 8082

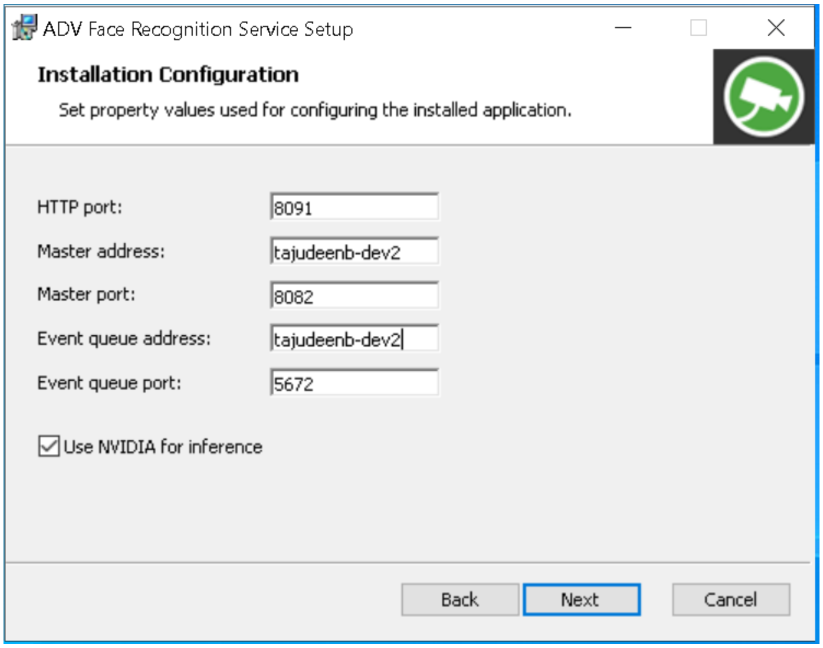
Event queue address: 127.0.0.1

Event queue port: 5672

☑ Use NVIDIA for inference

*Figure 1 FR Installed on local machine*

a. If Face Recognition Service is installed on other machine specifying the name or IP address of the machine using the Face Recognition Service (in this case, tajudeenb-dev2 is using Face Recognition Service from some other machine)



ADV Face Recognition Service Setup

**Installation Configuration**

Set property values used for configuring the installed application.

HTTP port: 8091

Master address: tajudeenb-dev2

Master port: 8082

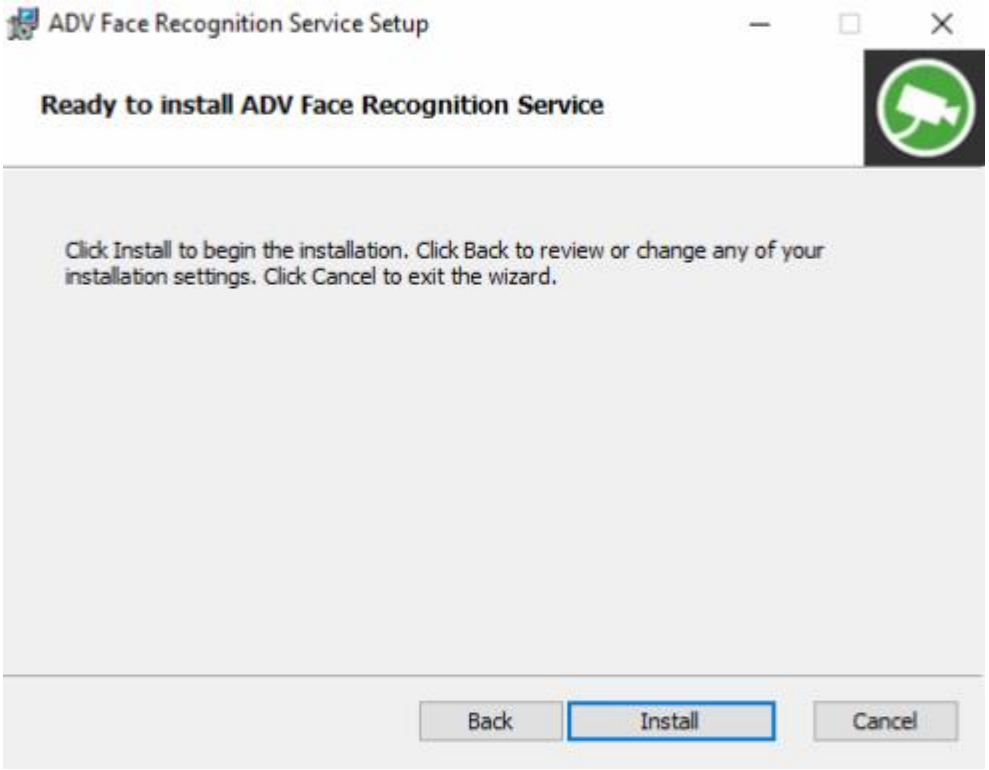
Event queue address: tajudeenb-dev2

Event queue port: 5672

☑ Use NVIDIA for inference

*Figure 2 FR Installed on other machine*

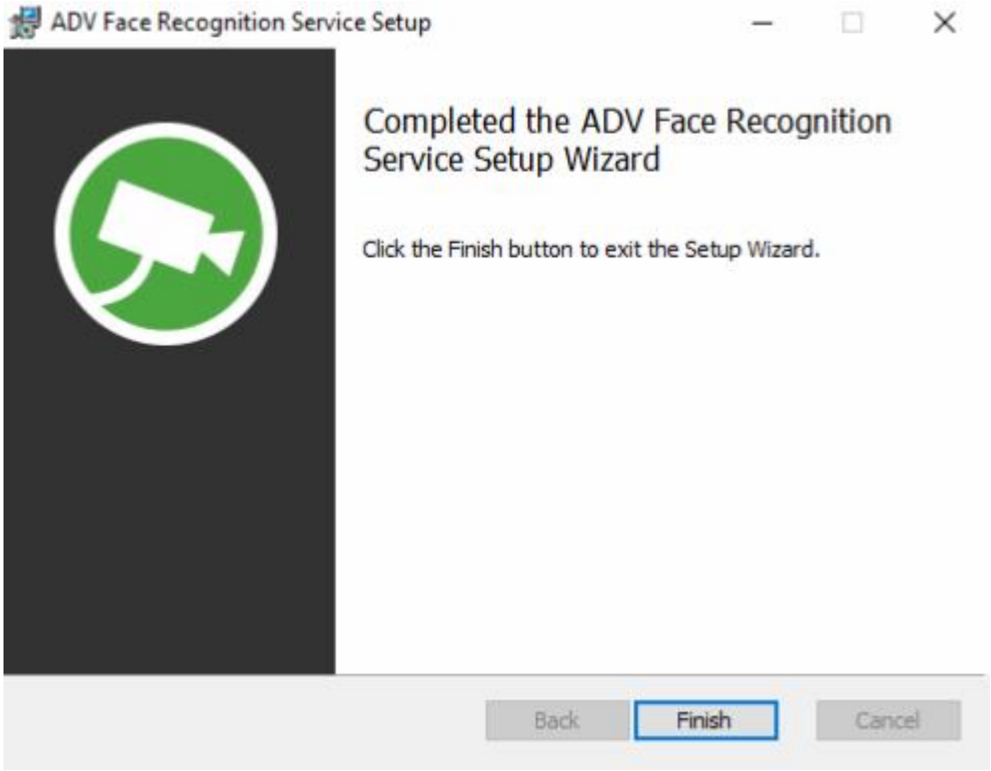
9. Click Install to continue and wait.
10. Installation will take some times until it finished.
11. Models creation can take up to 30 minutes. This depends how powerful graphics card is in use.



Ready to install ADV Face Recognition Service

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

12. Click Finish to continue.

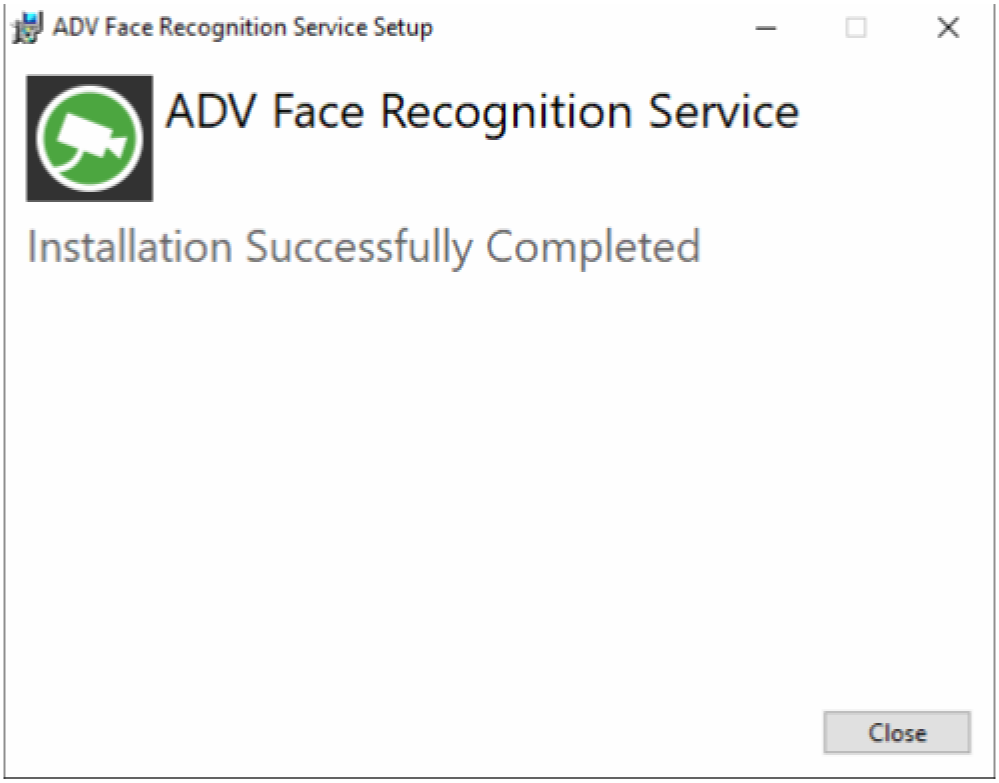


ADV Face Recognition Service Setup

Completed the ADV Face Recognition Service Setup Wizard

Click the Finish button to exit the Setup Wizard.

13. Click Close to close installation.



ADV Face Recognition Service

Installation Successfully Completed

14. Now Face Recognition Service is installed to server and ready to use.
    a. Face Recognition Service send details to VMS Master server and you can configure service via System Manager.

## 2.3 Privacy masks

If any client privacy masks are defined for the camera, the FR service draws privacy masks to input images before inference.

- No face can be detected inside the privacy zone.
- Thumbnail images have privacy zones.

## 2.4 FR PROCESSING, EVENTS, AND DETECTION

### 2.4.1 Devices

Face recognition processing can be done using different hardware. Supported hardware is CPU, Intel GPU, Nvidia GPU, and MAIC (Mirasys AI Card).

### 2.4.2 FR events

Live FR events are shown in the Smart Recognition plugin in Spotter. FR events can be searched using the Smart Search plugin in Spotter.

### 2.4.3 Detected face visualization

Detected faces can be visualized in Spotter using the VCA visualization plugin (Highlight menu in camera toolbar).

## 2.5 FR ALARM TRIGGERS AND CONFIGURATION

### 2.5.1 Alarm triggers

An alarm trigger on the VMS server can be created for each identity list that is configured in List Management settings.

### 2.5.2 FR configuration

FR service can be configured in the System Manager application on the FR settings tab in the **Camera Settings** window.

The FR settings contain information about camera video streams processed by the service. Each stream setting is related to the camera and stream on the recorder. Each FR service can have its own set of limits.

## 2.6 FR PERFORMANCE

### 2.6.1 Test machine

- 12th Gen Intel(R) Core(TM) i9-12900KF 3.19 GHz
- NVIDIA GeForce RTX 3080 Ti
- Intel UHD Graphics 750

### 2.6.2 How test is done

- FR service is restarted before starting to do test.
    - This free memory from test machine.
- HW video decoding used.
- Best FPS value is marked as orange.
    - This is best value without frame skipping.

### 2.6.3 Test results

#### *2.6.3.1 CPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | 21 | 15 % |  | 3 % |  | 2.8 GB |
| 2 | 640 x 480 | 42 | 42 | 27 % |  | 6 % |  | 3.1 GB |
| 3 | 640 x 480 | 63 | 63 | 46 % |  | 8 % |  | 3.5 GB |
| 4 | 640 x 480 | 84 | **83** | 53 % |  | 11 % |  | 3.9 GB |
| 5 | 640 x 480 | 105 | 85 | 50 % |  | 11 % |  | 4.2 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | 21 | 16 % |  | 9 % | 2.5 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 2 | 1920 x 1080 | 42 | 42 | 36 % | | 18 % | 3.1 GB |
| 3 | 1920 x 1080 | 63 | **63** | 51 % | | 27 % | 3.4 GB |
| 4 | 1920 x 1080 | 80 | 80 | 62 % | | 33 % | 4.9 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | 21 | 17 % | | 33 % | 2.9 GB |
| 2 | 3840 x 2160 | 42 | 42 | 35 % | | 60 % | 3.7 GB |
| 3 | 3840 x 2160 | 63 | **63** | 53 % | | 64 % | 4.4 GB |
| 4 | 3840 x 2160 | 80 | 80 | 70 % | | 60 % | 9.2 GB |

### *2.6.3.2 NVIDIA GPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | 21 | 2 % | 25 % | 3 % | | 2.4 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 2 | 640 x 480 | 42 | 42 | 4 % | 30 % | 5 % | | 2.6 GB |
| 3 | 640 x 480 | 63 | 63 | 4 % | 30 % | 5 % | | 2.8 GB |
| 4 | 640 x 480 | 84 | 84 | 4 % | 30 % | 5 % | | 3.0 GB |
| 5 | 640 x 480 | 105 | 105 | 5 % | 30 % | 7 % | | 3.2 GB |
| 6 | 640 x 480 | 126 | 126 | 6 % | 35 % | 8 % | | 3.3 GB |
| 7 | 640 x 480 | 147 | 147 | 7 % | 38 % | 9 % | | 3.4 GB |
| 8 | 640 x 480 | 168 | 168 | 8 % | 90 % | 9 % | | 3.6 GB |
| 9 | 640 x 480 | 189 | 189 | 8 % | 96 % | 9 % | | 3.7 GB |
| 10 | 640 x 480 | 210 | 210 | 11 % | 96 % | 9 % | | 3.8 GB |
| 11 | 640 x 480 | 231 | 231 | 11 % | 96 % | 9 % | | 3.9 GB |
| 12 | 640 x 480 | 252 | **252** | 14 % | 96 % | 10 % | | 4.0 GB |
| 13 | 640 x 480 | 273 | 243 | 15 % | 92 % | 10 % | | 4.8 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | 21 | 4 % | 25 % | 9 % | | 2.4 GB |
| 2 | 1920 x 1080 | 42 | 42 | 3 % | 28 % | 15 % | | 2.7 GB |
| 3 | 1920 x 1080 | 63 | 63 | 5 % | 28 % | 14 % | | 2.9 GB |
| 4 | 1920 x 1080 | 84 | 84 | 6 % | 30 % | 17 % | | 3.2 GB |
| 5 | 1920 x 1080 | 105 | 105 | 7 % | 35 % | 22 % | | 3.3 GB |
| 6 | 1920 x 1080 | 126 | 126 | 10 % | 35 % | 24 % | | 3.6 GB |
| 7 | 1920 x 1080 | 147 | 147 | 11 % | 95 % | 25 % | | 3.7 GB |
| 8 | 1920 x 1080 | 168 | 168 | 13 % | 95 % | 25 % | | 3.9 GB |
| 9 | 1920 x 1080 | 189 | 189 | 17 % | 95 % | 25 % | | 4.0 GB |
| 10 | 1920 x 1080 | 210 | 210 | 19 % | 95 % | 28 % | | 4.2 GB |
| 11 | 1920 x 1080 | 231 | 231 | 22 % | 95 % | 31 % | | 4.4 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 12 | 1920 x 1080 | 252 | **252** | 26 % | 95 % | 34 % | | 5.6 GB |
| 13 | 1920 x 1080 | 249 | 249 | 29 % | 93 % | 33 % | | 6.5 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | 21 | 3 % | 32 % | 32 % | | 2.7 GB |
| 2 | 3840 x 2160 | 42 | 42 | 7 % | 37 % | 37 % | | 3.1 GB |
| 3 | 3840 x 2160 | 63 | 63 | 10 % | 44 % | 44 % | | 3.5 GB |
| 4 | 3840 x 2160 | 84 | 84 | 13 % | 60 % | 60 % | | 3.8 GB |
| 5 | 3840 x 2160 | 105 | 105 | 21 % | 95 % | 60 % | | 4.2 GB |
| 6 | 3840 x 2160 | 126 | 126 | 22 % | 95 % | 61 % | | 4.5 GB |
| 7 | 3840 x 2160 | 147 | 147 | 28 % | 97 % | 71 % | | 4.8 GB |
| 8 | 3840 x 2160 | 168 | 168 | 30 % | 96 % | 81 % | | 5.1 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 9 | 3840 x 2160 | 189 | 189 | 35 % | 95 % | 91 % | | 5.9 GB |
| 10 | 3840 x 2160 | 210 | **210** | 37 % | 100 % | 100 % | | 14.7 GB |
| 11 | 3840 x 2160 | 209 | 209 | 40 % | 100 % | 100 % | | 14.9 GB |

### *2.6.3.3 INTEL GPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | **21** | 7 % | | 3 % | 52 % | 1.8 GB |
| 2 | 640 x 480 | 40 | 40 | 20 % | | 7 % | 100 % | 2.6 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | **21** | 8 % | | 5 % | 55 % | 1.8 GB1 |
| 2 | 1920 x 1080 | 39 | 39 | 22 % | | 15 % | 99 % | 3.2 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | **21** | 10 % | | 30 % | 55 % | 2.0 GB |
| 2 | 3840 x 2160 | 39 | 39 | 27 % | | 60 % | 98 % | 5.2 GB |

These tests are indicative and may not be directly applicable to production systems.

# 3 Mirasys License Plate Recognition (LPR)

## 3.1 License Plate Recognition Introduction

License Plate Recognition (LPR) is used to identify a car using its license plate. It is used in the VMS System to get events when license plates are detected from selected video streams and to detect when specific cars are seen in the video. Together with the Mirasys List Management, this allows you to, for example, create an automatic detection system of cars' access to the parking hall.

The LPR service receives video streams, processes images, detects license plates, and sends notifications with detection data to List Management (LM) service for identity and list matching.

License Plate Recognition service has a separate installer, so it can execute on a separate server or on some VMS server.

On licensing side there is need to add RTSP Streaming Server feature to Management Server and all those servers where is plan to use SmartLPR feature. On Management Server side there is need to be add wanted amount of channels of SmartLPR feature.

## 3.2 LPR Service Installation

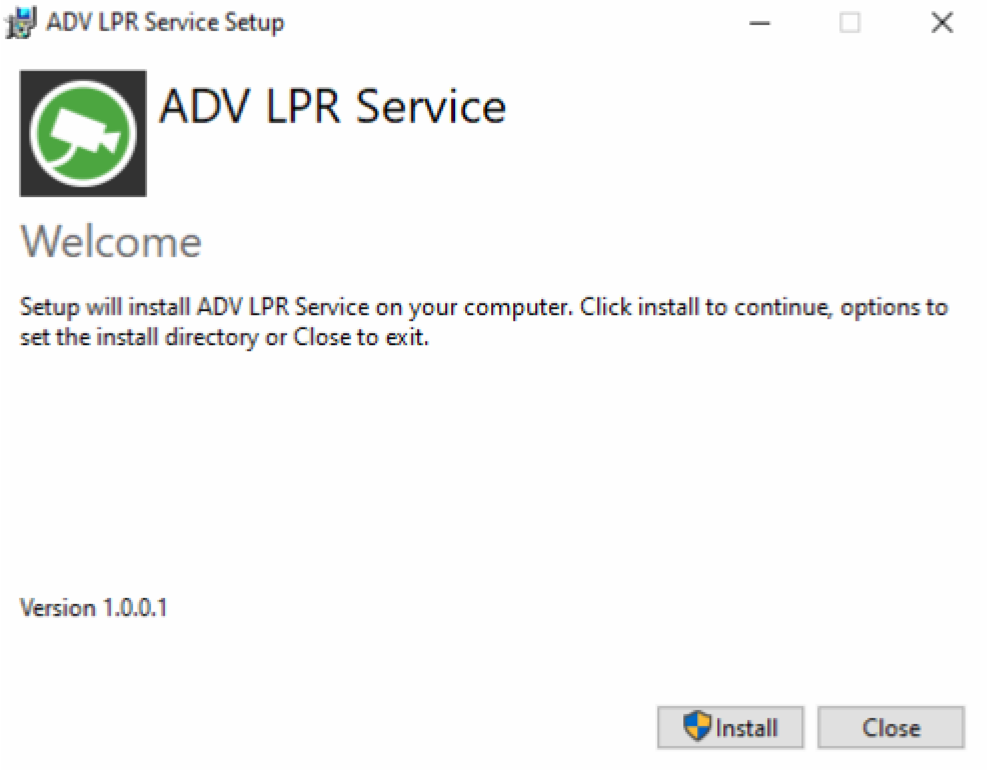
Smart services can be used together with the VCA Deep Learning feature. In this case, you should note that you are using the latest NVIDIA drivers, and not the ones that come with the CUDA Toolkit package. More information can be found here.

### 3.2.1 Requirements

- Administrator rights
- List Management Service installed
- License Plate Recognition license on VMS server

### 3.2.2 Installation

1. Download latest version from Extranet.
2. Unzip this example to C:\temp folder.
3. Start installation double clicking installation file.
4. Click Install to continue.



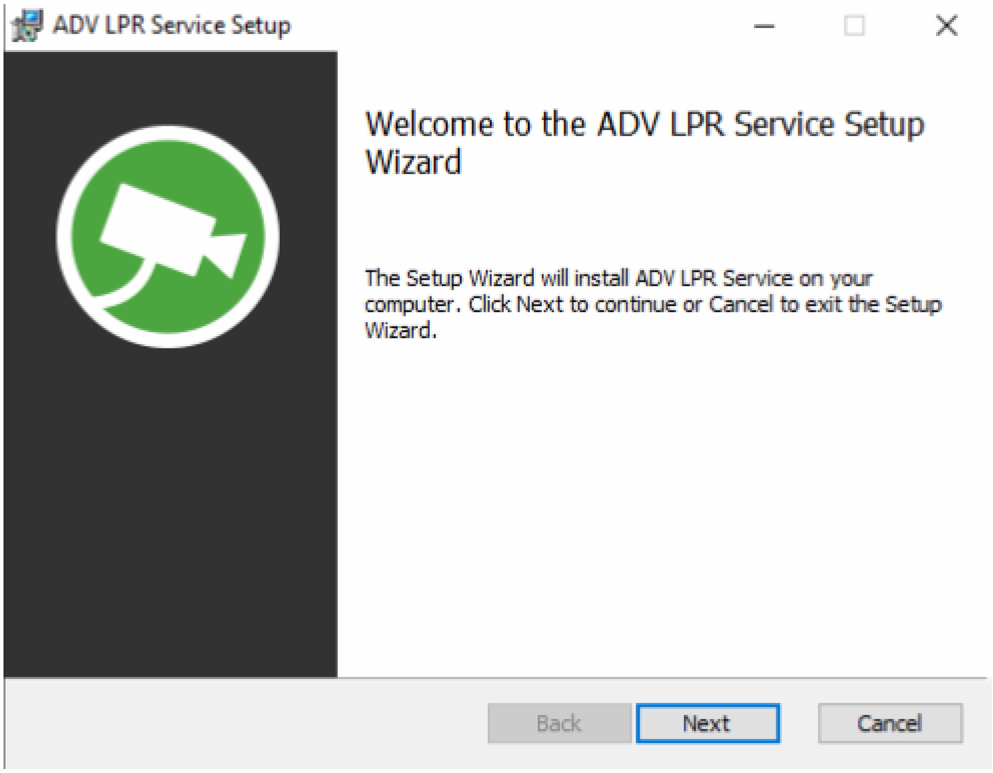
ADV LPR Service

Welcome

Setup will install ADV LPR Service on your computer. Click install to continue, options to set the install directory or Close to exit.

Version 1.0.0.1

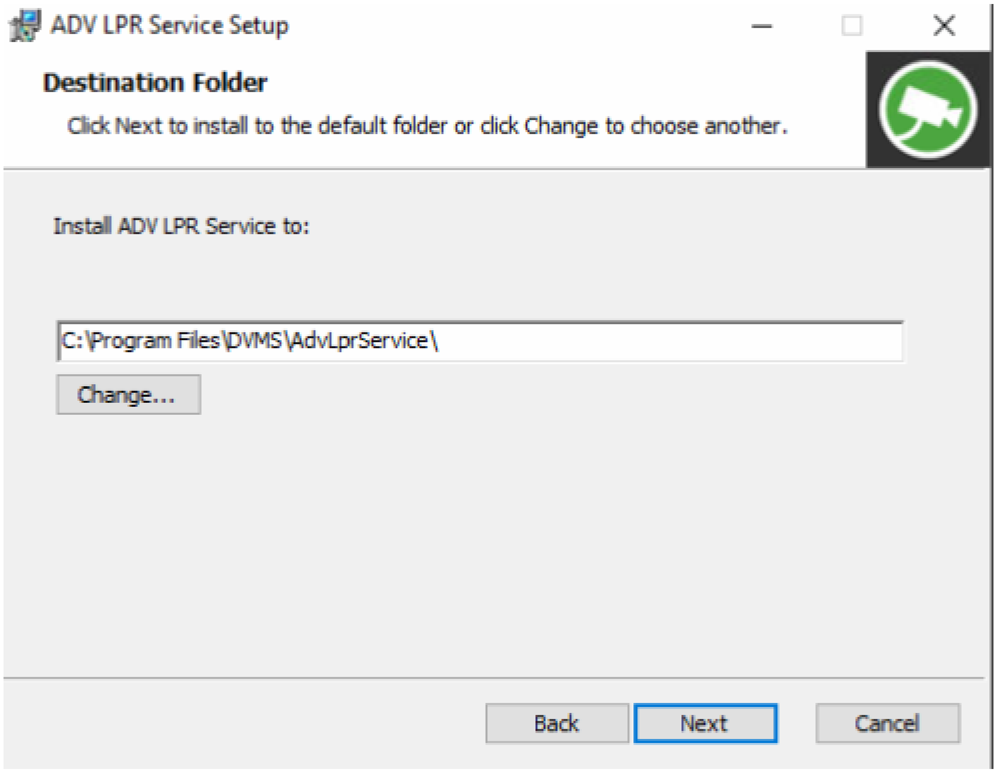
5. Click Next to continue.



Welcome to the ADV LPR Service Setup Wizard

The Setup Wizard will install ADV LPR Service on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

6. Change installation location if needed, if not then click Next to continue.



ADV LPR Service Setup

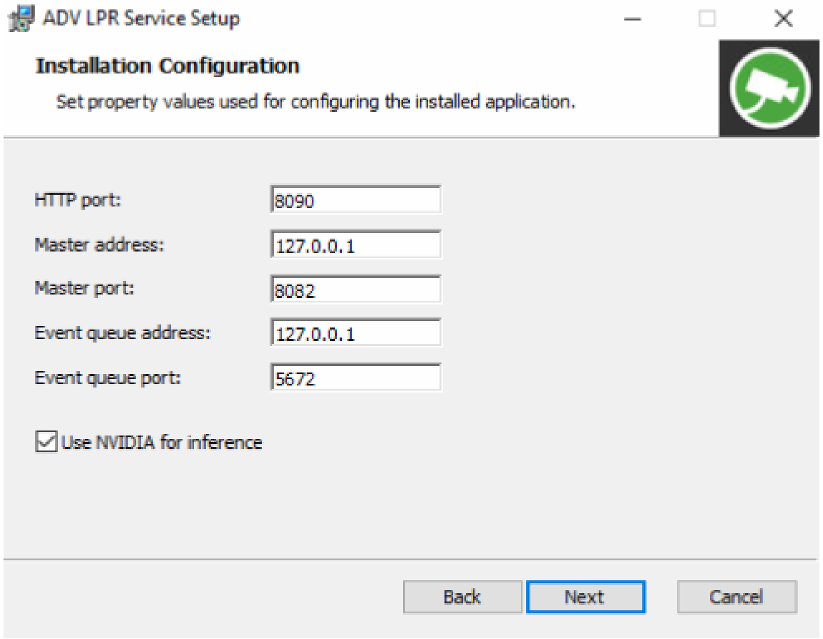
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install ADV LPR Service to:

C:\Program Files\DVMS\AdvLprService\

7. Change ports and addresses if needed.
    a. If you example install Face Recognition service to other machine than VMS Master, you need change Master address to correct one.
    b. Same apply for Event queue address. Replace this address with that server address where List Management Service is installed.
    c. If you have Nvidia graphics card installed to server, you can keep Use NVIDIA for inference enabled. This create Nvidia models to use graphics card.
8. Click Next to continue.



ADV LPR Service Setup

**Installation Configuration**

Set property values used for configuring the installed application.

HTTP port: 8090

Master address: 127.0.0.1

Master port: 8082

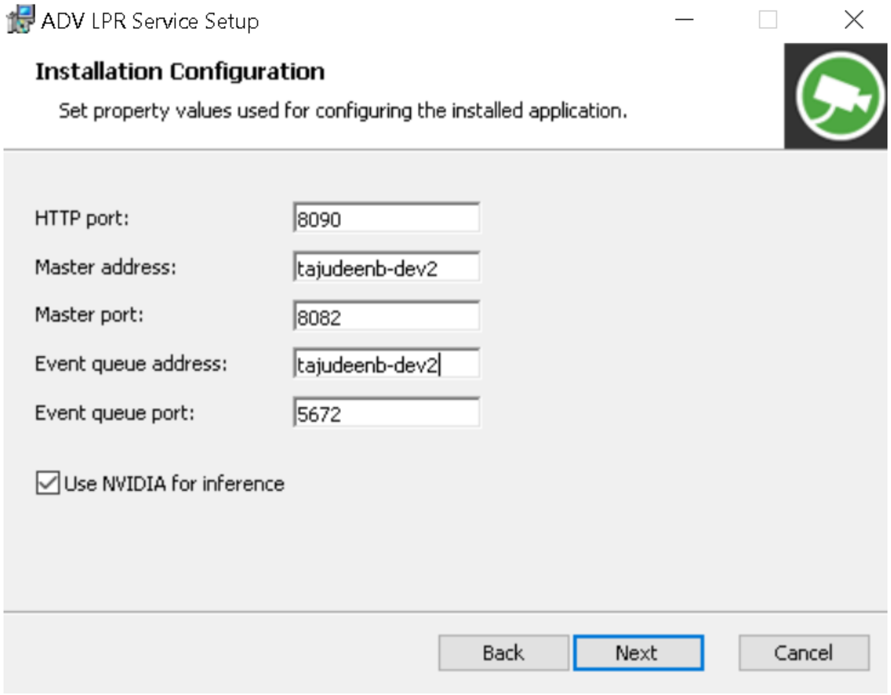
Event queue address: 127.0.0.1

Event queue port: 5672

☑ Use NVIDIA for inference

*Figure 3 LPR Installed on local machine*

a. If License Plate Recognition Service is installed on some other machine specifying the machine name or IP address using the service (in this case, tajudeenb-dev2 is using License Plate Recognition Service from some other machine)



**Installation Configuration**

Set property values used for configuring the installed application.

HTTP port: 8090

Master address: tajudeenb-dev2

Master port: 8082

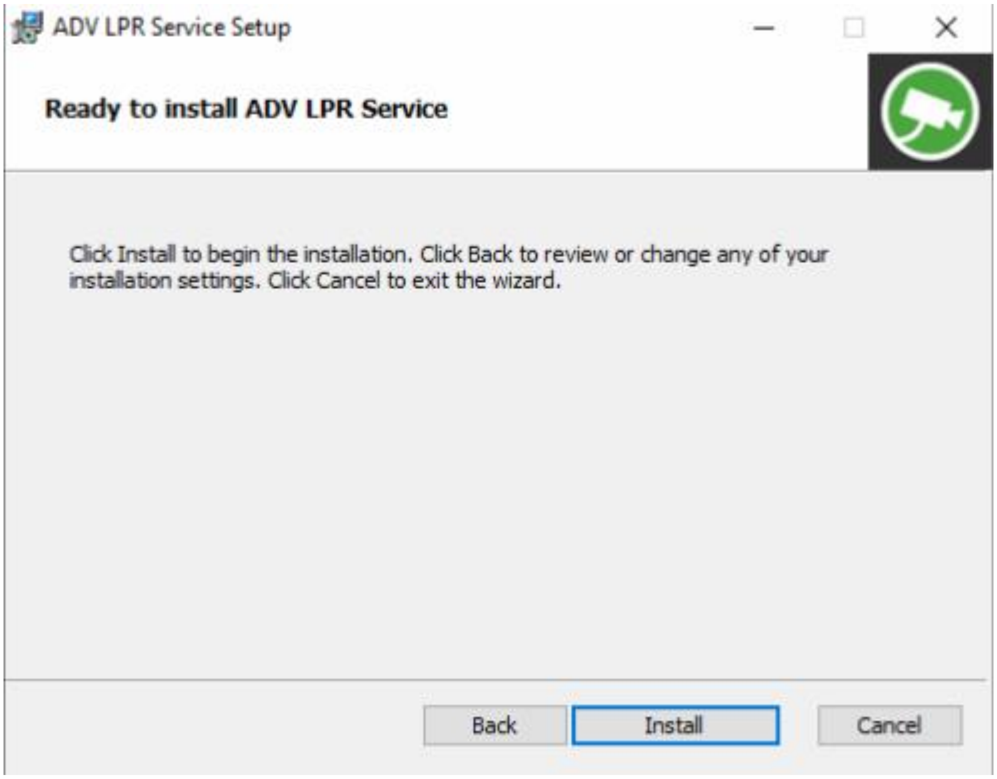
Event queue address: tajudeenb-dev2

Event queue port: 5672

☑ Use NVIDIA for inference

*Figure 4 LPR service installed on some other machine*

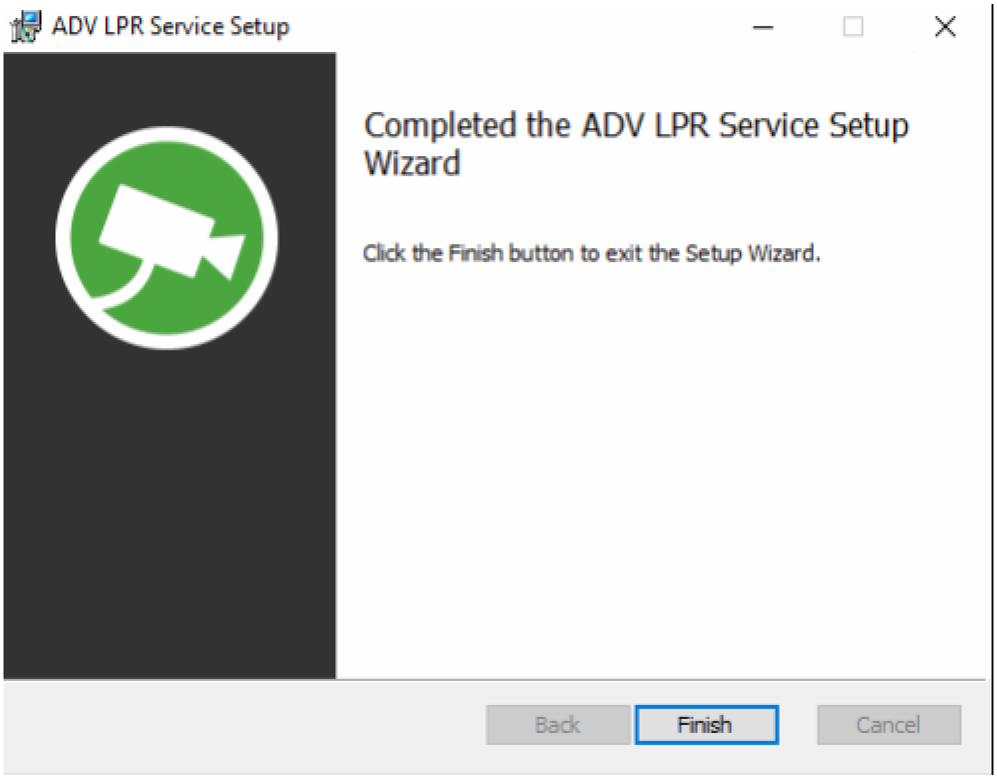
9. Click Install to continue and wait.
    a. Installation will take some times until it finished.
    b. Models creation can take up to 30 minutes. This depends how powerful graphics card is in use.



Ready to install ADV LPR Service

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

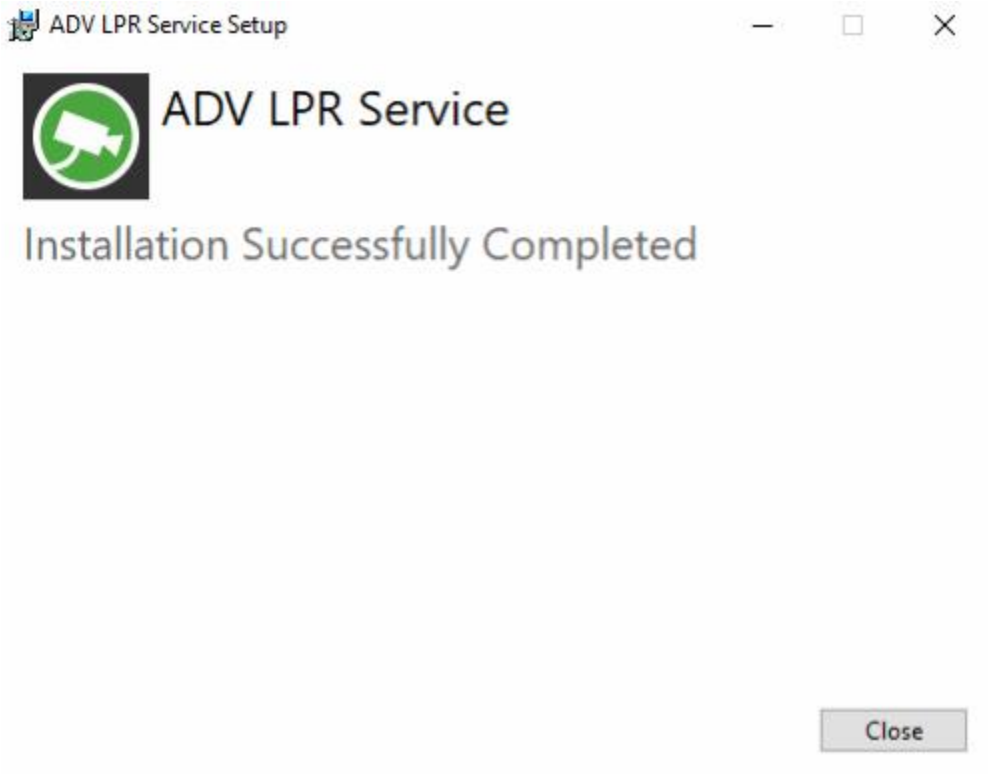
10. Click Finish to continue.



Completed the ADV LPR Service Setup Wizard

Click the Finish button to exit the Setup Wizard.

11. Click Close to close installation.



ADV LPR Service Setup

ADV LPR Service

Installation Successfully Completed

12. Now License Plate Recognition Service is installed to server and ready to use.
    a. License Plate Recognition Service send details to VMS Master server and you can configure service via System Manager.

## 3.3 LPR Privacy masks

If any client privacy masks are defined for the camera, then the LPR service draws privacy masks to input images before doing inference.

- No license plate can be detected inside the privacy zone.
- Thumbnail images have privacy zones.

## 3.4 Country Detection

Country detection is optional, but in some countries, it is recommended to be used: plate number detection accuracy can be improved when the country is known.

### 3.4.1 Plate number country detection

Plate number detection is available for Eurasia and the Americas.

Country detection is optional. It is useful in countries like Finland, where the letters I and O are the same as numbers 1 and 0. If the country is recognized with high confidence, then country-specific rules can be used to improve plate number detection accuracy.

### 3.4.2 License plate types

If country detection is enabled, then license plate type can sometimes be detected. License plate type can be undefined or one of these:

- antique
- diplomatic
- export
- military
- provisional
- rental
- taxi
- test
- work

## 3.5 Supported countries in Eurasia (LPR)

### 3.5.1 Area codes

In some countries, license plates have area codes. If country detection is enabled, then also area code is detected for the following countries:

- Austria
- Germany
- Romania
- Slovenia
- Switzerland

In a special case, a region inside a country can be detected. For example, Åland Islands has its own plate styles, different from those used in other parts of Finland.

> Please note that accuracy vary from country to country.
> A plate from **unsupported** countries could be detected as one of the countries listed below.
> For example some Tajikistan plates can be detected as Kazakhstan plates.

### 3.5.2 List of supported countries in Eurasia

- Albania
- Andorra
- Armenia
- Austria
- Azerbaijan
- Belarus
- Belgium
- Bosnia and Herzegovina
- Bulgaria
- Croatia
- Cyprus
- Czech Republic

- Denmark
- Estonia
- Finland (including Åland Islands)
- France
- Georgia
- Germany
- Gibraltar
- Greece
- Hungary
- Iceland
- Ireland
- Isle of Man
- Italy
- Kazakhstan
- Latvia
- Liechtenstein
- Lithuania
- Luxembourg
- Malta
- Moldova
- Monaco
- Montenegro

- Netherlands
- North Macedonia
- Norway
- Poland
- Portugal
- Romania
- Russia
- San Marino
- Serbia
- Slovakia
- Slovenia
- Spain
- Sweden
- Switzerland
- Turkey
- Ukraine
- United Kingdom
- Vatican

## 3.6 Supported countries in the Americas (LPR)

Please note that accuracy vary from country to country.
A plate from **unsupported** countries could be detected as one of the countries listed below.

### 3.6.1 Countries and states

The countries/states listed below are supported.

- Argentina
- Bolivia
- Brazil (old and new plate styles)
- Canada
  - Alberta
  - British Columbia
  - Manitoba
  - Ontario
  - Quebec
  - Saskatchewan
- Chile
- Colombia
- Mexico
- Paraguay
- Peru
- United States
  - Alabama
  - Alaska
  - Arizona
  - Arkansas
  - California

- Colorado
- Connecticut
- Delaware
- District of Columbia
- Florida
- Georgia
- Hawaii
- Idaho
- Illinois
- Indiana
- Iowa
- Kansas
- Kentucky
- Louisiana
- Maine
- Maryland
- Massachusetts
- Michigan
- Minnesota
- Mississippi
- Missouri
- Montana

- Nebraska
- Nevada
- New Hampshire
- New Jersey
- New Mexico
- New York
- North Carolina
- North Dakota
- Ohio
- Oklahoma
- Oregon
- Pennsylvania
- Rhode Island
- South Carolina
- South Dakota
- Tennessee
- Texas
- Utah
- Vermont
- Virginia
- Washington
- West Virginia

  - Wisconsin
  - Wyoming
- Uruguay
- Venezuela

## 3.7 LPR PROCESSING, EVENTS, AND DETECTION

### 3.7.1 Devices

License plate recognition processing can be done using different hardware. Supported hardware is CPU, Intel GPU, Nvidia GPU, and MAIC (Mirasys AI Card).

### 3.7.2 LPR events

Live LPR events are shown in the Smart Recognition plugin in Spotter. LPR events can be searched using the Smart Search Plugin in Spotter.

### 3.7.3 Detected license plate visualization

Detected license plates can be visualized in Spotter using the VCA visualization plugin (Highlight menu in camera toolbar).

## 3.8 LPR ALARM TRIGGERS AND CONFIGURATION

### 3.8.1 Alarm triggers

An alarm trigger on the VMS server can be created for each identity list that is configured in List Management settings.

### 3.8.2 LPR configuration

LPR service can be configured in the System Manager application on the LPR settings tab in the **Camera Settings** window.

The LPR settings contain information about camera video streams processed by the service. Each stream setting is related to the camera and stream on the recorder. Each LPR service can have its own set of limits.

## 3.9 LPR Performance

### 3.9.1 Test machine

- 12th Gen Intel(R) Core(TM) i9-12900KF 3.19 GHz
- NVIDIA GeForce RTX 3080 Ti
- Intel UHD Graphics 750

### 3.9.2 How test is done

- LPR service is restarted before starting to do test.
    - This free memory from test machine.
- HW video decoding used.
- Best FPS value is marked as orange.
    - This is best value without frame skipping.

### 3.9.3 Test results

#### *3.9.3.1 CPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | 21 | 15 % | | 3 % | | 2.8 GB |
| 2 | 640 x 480 | 42 | 42 | 27 % | | 6 % | | 3.1 GB |
| 3 | 640 x 480 | 63 | 63 | 46 % | | 8 % | | 3.5 GB |
| 4 | 640 x 480 | 84 | **83** | 53 % | | 11 % | | 3.9 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 5 | 640 x 480 | 105 | 85 | 50 % | | 11 % | | 4.2 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | 21 | 16 % | | 9 % | 2.5 GB |
| 2 | 1920 x 1080 | 42 | 42 | 36 % | | 18 % | 3.1 GB |
| 3 | 1920 x 1080 | 63 | **63** | 51 % | | 27 % | 3.4 GB |
| 4 | 1920 x 1080 | 80 | 80 | 62 % | | 33 % | 4.9 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | 21 | 17 % | | 33 % | 2.9 GB |
| 2 | 3840 x 2160 | 42 | 42 | 35 % | | 60 % | 3.7 GB |
| 3 | 3840 x 2160 | 63 | **63** | 53 % | | 64 % | 4.4 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | Process memory |
|---|---|---|---|---|---|---|---|
| 4 | 3840 x 2160 | 80 | 80 | 70 % | | 60 % | 9.2 GB |

### *3.9.3.2 NVIDIA GPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | 21 | 2 % | 25 % | 3 % | | 2.4 GB |
| 2 | 640 x 480 | 42 | 42 | 4 % | 30 % | 5 % | | 2.6 GB |
| 3 | 640 x 480 | 63 | 63 | 4 % | 30 % | 5 % | | 2.8 GB |
| 4 | 640 x 480 | 84 | 84 | 4 % | 30 % | 5 % | | 3.0 GB |
| 5 | 640 x 480 | 105 | 105 | 5 % | 30 % | 7 % | | 3.2 GB |
| 6 | 640 x 480 | 126 | 126 | 6 % | 35 % | 8 % | | 3.3 GB |
| 7 | 640 x 480 | 147 | 147 | 7 % | 38 % | 9 % | | 3.4 GB |
| 8 | 640 x 480 | 168 | 168 | 8 % | 90 % | 9 % | | 3.6 GB |
| 9 | 640 x 480 | 189 | 189 | 8 % | 96 % | 9 % | | 3.7 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 10 | 640 x 480 | 210 | 210 | 11 % | 96 % | 9 % | | 3.8 GB |
| 11 | 640 x 480 | 231 | 231 | 11 % | 96 % | 9 % | | 3.9 GB |
| 12 | 640 x 480 | 252 | **252** | 14 % | 96 % | 10 % | | 4.0 GB |
| 13 | 640 x 480 | 273 | 243 | 15 % | 92 % | 10 % | | 4.8 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | 21 | 4 % | 25 % | 9 % | | 2.4 GB |
| 2 | 1920 x 1080 | 42 | 42 | 3 % | 28 % | 15 % | | 2.7 GB |
| 3 | 1920 x 1080 | 63 | 63 | 5 % | 28 % | 14 % | | 2.9 GB |
| 4 | 1920 x 1080 | 84 | 84 | 6 % | 30 % | 17 % | | 3.2 GB |
| 5 | 1920 x 1080 | 105 | 105 | 7 % | 35 % | 22 % | | 3.3 GB |
| 6 | 1920 x 1080 | 126 | 126 | 10 % | 35 % | 24 % | | 3.6 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 7 | 1920 x 1080 | 147 | 147 | 11 % | 95 % | 25 % | | 3.7 GB |
| 8 | 1920 x 1080 | 168 | 168 | 13 % | 95 % | 25 % | | 3.9 GB |
| 9 | 1920 x 1080 | 189 | 189 | 17 % | 95 % | 25 % | | 4.0 GB |
| 10 | 1920 x 1080 | 210 | 210 | 19 % | 95 % | 28 % | | 4.2 GB |
| 11 | 1920 x 1080 | 231 | 231 | 22 % | 95 % | 31 % | | 4.4 GB |
| 12 | 1920 x 1080 | 252 | **252** | 26 % | 95 % | 34 % | | 5.6 GB |
| 13 | 1920 x 1080 | 249 | 249 | 29 % | 93 % | 33 % | | 6.5 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | 21 | 3 % | 32 % | 32 % | | 2.7 GB |
| 2 | 3840 x 2160 | 42 | 42 | 7 % | 37 % | 37 % | | 3.1 GB |
| 3 | 3840 x 2160 | 63 | 63 | 10 % | 44 % | 44 % | | 3.5 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 4 | 3840 x 2160 | 84 | 84 | 13 % | 60 % | 60 % | | 3.8 GB |
| 5 | 3840 x 2160 | 105 | 105 | 21 % | 95 % | 60 % | | 4.2 GB |
| 6 | 3840 x 2160 | 126 | 126 | 22 % | 95 % | 61 % | | 4.5 GB |
| 7 | 3840 x 2160 | 147 | 147 | 28 % | 97 % | 71 % | | 4.8 GB |
| 8 | 3840 x 2160 | 168 | 168 | 30 % | 96 % | 81 % | | 5.1 GB |
| 9 | 3840 x 2160 | 189 | 189 | 35 % | 95 % | 91 % | | 5.9 GB |
| 10 | 3840 x 2160 | 210 | **210** | 37 % | 100 % | 100 % | | 14.7 GB |
| 11 | 3840 x 2160 | 209 | 209 | 40 % | 100 % | 100 % | | 14.9 GB |

### *3.9.3.3 INTEL GPU*

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 640 x 480 | 21 | **21** | 7 % | | 3 % | 52 % | 1.8 GB |
| 2 | 640 x 480 | 40 | 40 | 20 % | | 7 % | 100 % | 2.6 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 1920 x 1080 | 21 | **21** | 8 % | | 5 % | 55 % | 1.8 GB1 |
| 2 | 1920 x 1080 | 39 | 39 | 22 % | | 15 % | 99 % | 3.2 GB |

| Streams | Size | Input FPS | Output FPS | Process CPU | NVIDIA GPU | NVIDIA decode | INTEL GPU | Process memory |
|---|---|---|---|---|---|---|---|---|
| 1 | 3840 x 2160 | 21 | **21** | 10 % | | 30 % | 55 % | 2.0 GB |
| 2 | 3840 x 2160 | 39 | 39 | 27 % | | 60 % | 98 % | 5.2 GB |

These tests are indicative and may not be directly applicable to production systems.

# 4 Easy LPR Guide

## 4.1 EASY LPR main features

- Live monitoring from the one camera at the same time
- Plate number search from the one camera at the same time
- Plate number list Management
  - Black list
  - White list
- Importing and exporting plate number lists

- Uploading plate number list to the cameras
- Digital output controlling based on:
  - Other plate detected
  - Black list plate detected
  - White list plate detected

Please check supported cameras from Supported IP Camera List.

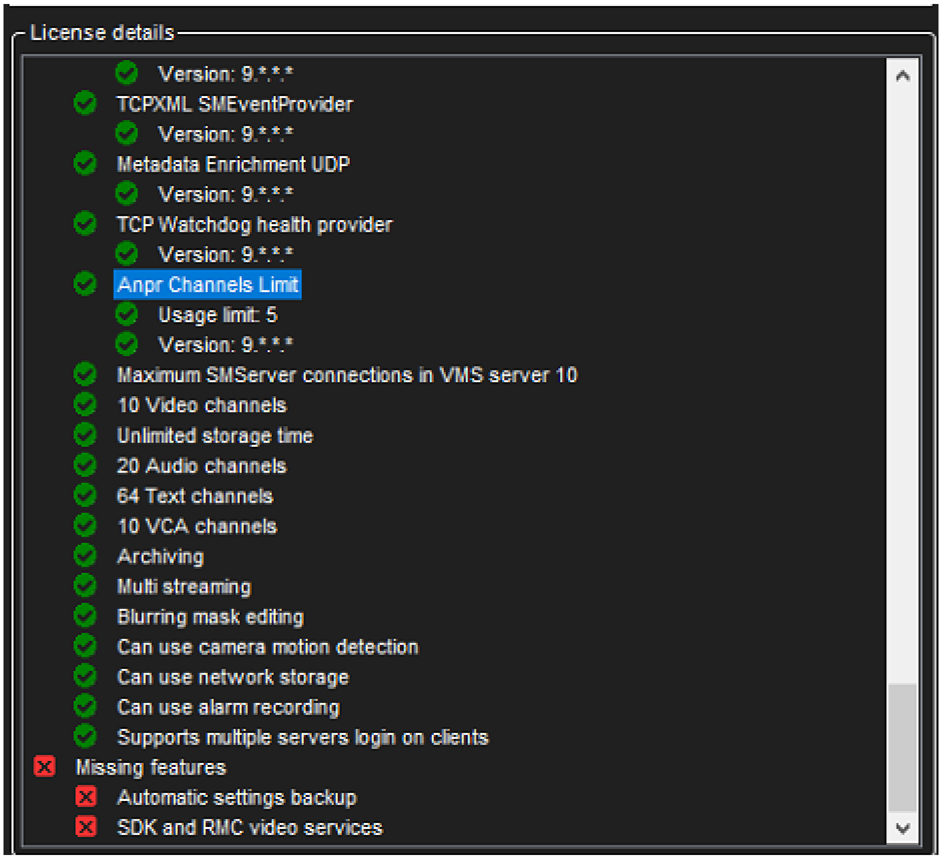
## 4.2 Easy LPR Configuration process

1. Configure LPR functionality to the used cameras. Please see the manufacturer website for more information
2. Check that license plates are correctly detected in the camera side
3. Add cameras to the Mirasys VMS
4. Check that Mirasys VMS license support LPR cameras
5. Enable Easy LPR

## 4.3 Easy LPR Licensing

Mirasys VMS server license defines how many ANPR channels can be added.

The feature name is **Anpr Channels limit** and controllable value name **Usage limit**



License details

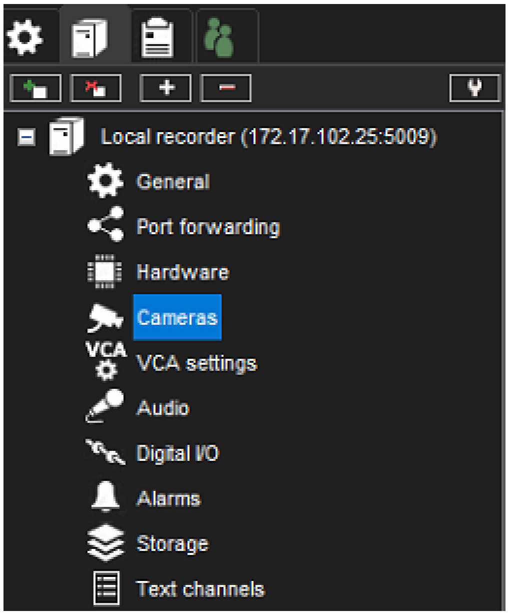
- ✓ Version: 9.*.*.*
- ✓ TCPXML SMEventProvider
  - ✓ Version: 9.*.*.*
- ✓ Metadata Enrichment UDP
  - ✓ Version: 9.*.*.*
- ✓ TCP Watchdog health provider
  - ✓ Version: 9.*.*.*
- ✓ Anpr Channels Limit
  - ✓ Usage limit: 5
  - ✓ Version: 9.*.*.*
- ✓ Maximum SMServer connections in VMS server 10
- ✓ 10 Video channels
- ✓ Unlimited storage time
- ✓ 20 Audio channels
- ✓ 64 Text channels
- ✓ 10 VCA channels
- ✓ Archiving
- ✓ Multi streaming
- ✓ Blurring mask editing
- ✓ Can use camera motion detection
- ✓ Can use network storage
- ✓ Can use alarm recording
- ✓ Supports multiple servers login on clients

☒ Missing features
- ☒ Automatic settings backup
- ☒ SDK and RMC video services

## 4.4 How to enable Easy LPR

1. Open **VMS servers**

2. Open **Cameras**

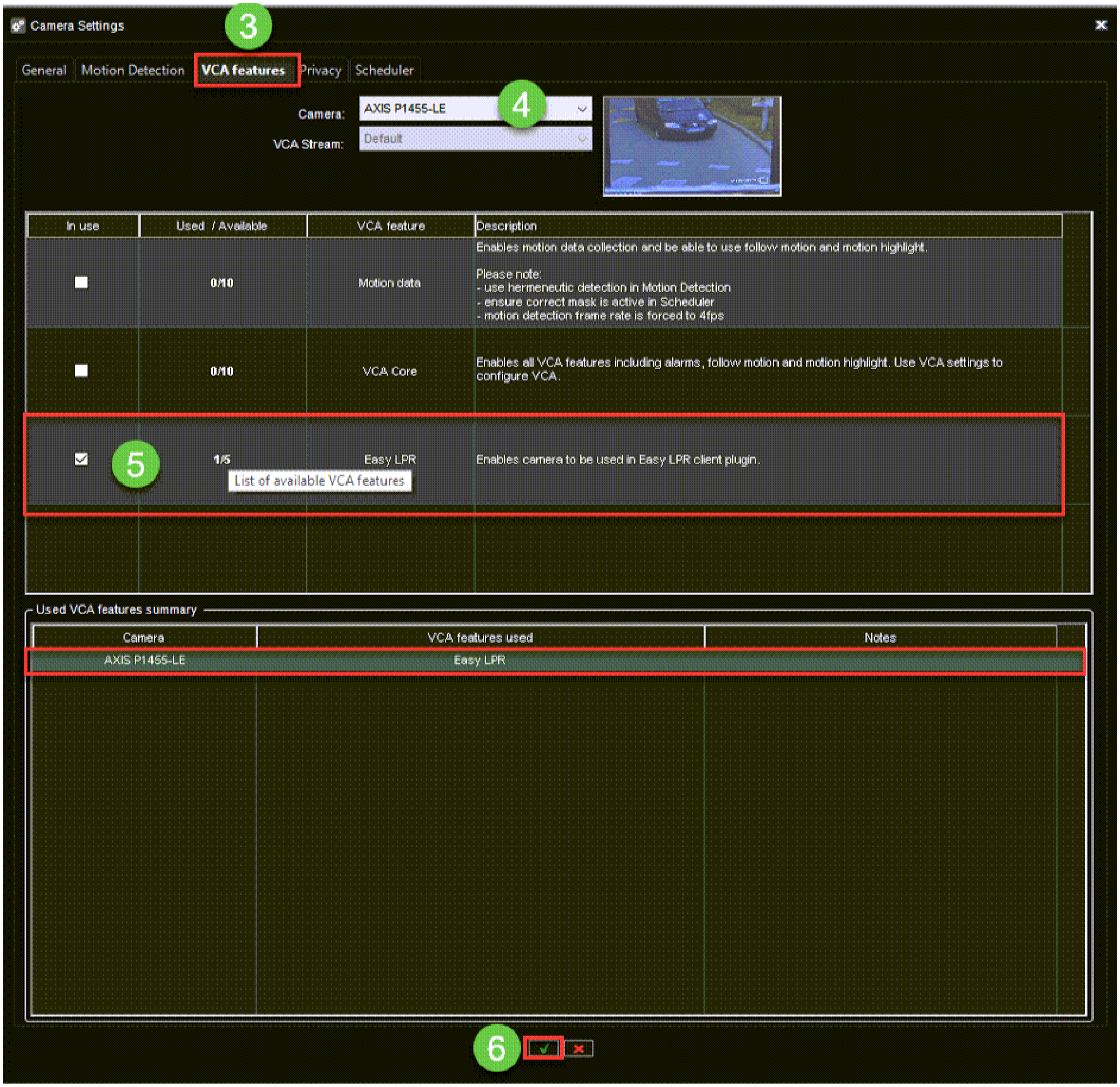


3. Click **VCA features**

4. Select LPR camera

5. Enable **Easy LPR**

6. Click **Save**



Camera: AXIS P1455-LE

VCA Stream: Default

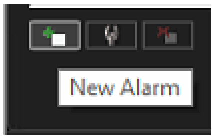
| In use | Used / Available | VCA feature | Description |
|---|---|---|---|
| □ | 0/10 | Motion data | Enables motion data collection and be able to use follow motion and motion highlight.<br>Please note:<br>- use hermeneutic detection in Motion Detection<br>- ensure correct mask is active in Scheduler<br>- motion detection frame rate is forced to 4fps |
| □ | 0/10 | VCA Core | Enables all VCA features including alarms, follow motion and motion highlight. Use VCA settings to configure VCA. |
| ☑ | 1/5 | Easy LPR | Enables camera to be used in Easy LPR client plugin. |

Used VCA features summary

| Camera | VCA features used | Notes |
|---|---|---|
| AXIS P1455-LE | Easy LPR | |

## 4.5 Create an Alarm from an Easy LPR event

1. Go to the **VMS Servers** tab
2. Open **Alarms**
3. Click **New Alarm**



4. Enter all needed information in the **General** tab

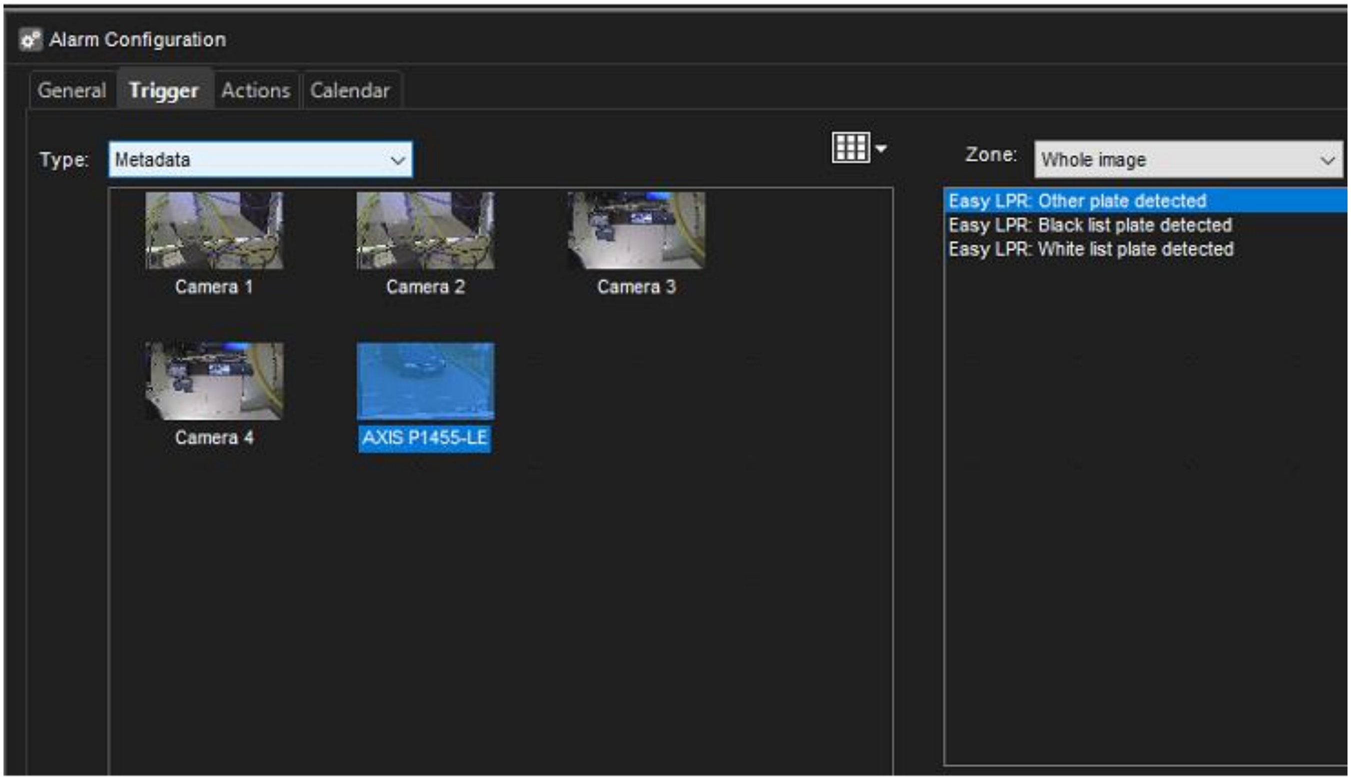
5. Open **Trigger** tab

6. Select trigger type **Metadata**

7. Select LPR camera

8. Select correct event:

- **Easy LPR: Other plate detected**
- **Easy LPR: Black list plate detected**
- **Easy LPR: White list plate detected**



Type: Metadata

Zone: Whole image

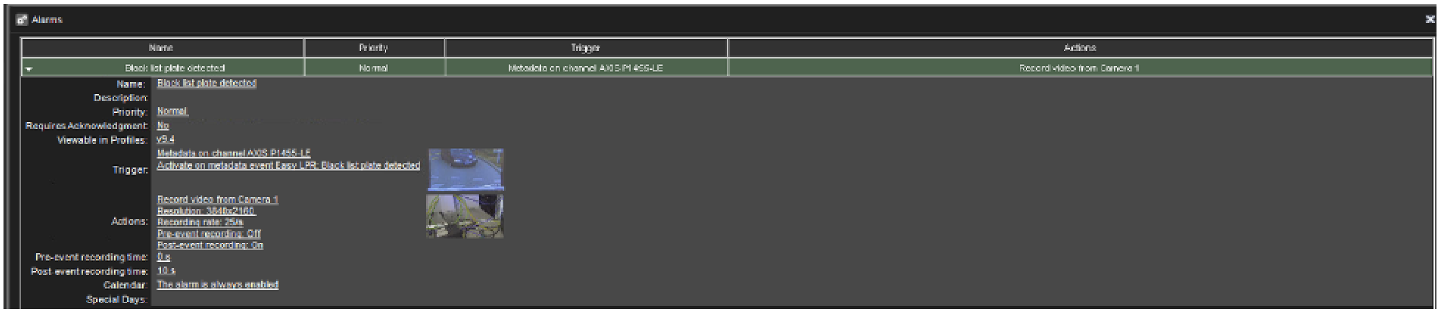
Easy LPR: Other plate detected
Easy LPR: Black list plate detected
Easy LPR: White list plate detected

9. Enter the actions of the alarms

10. Set calendar

11. Check overall view of the alarm

12. Click **OK** to confirm an alarm creation



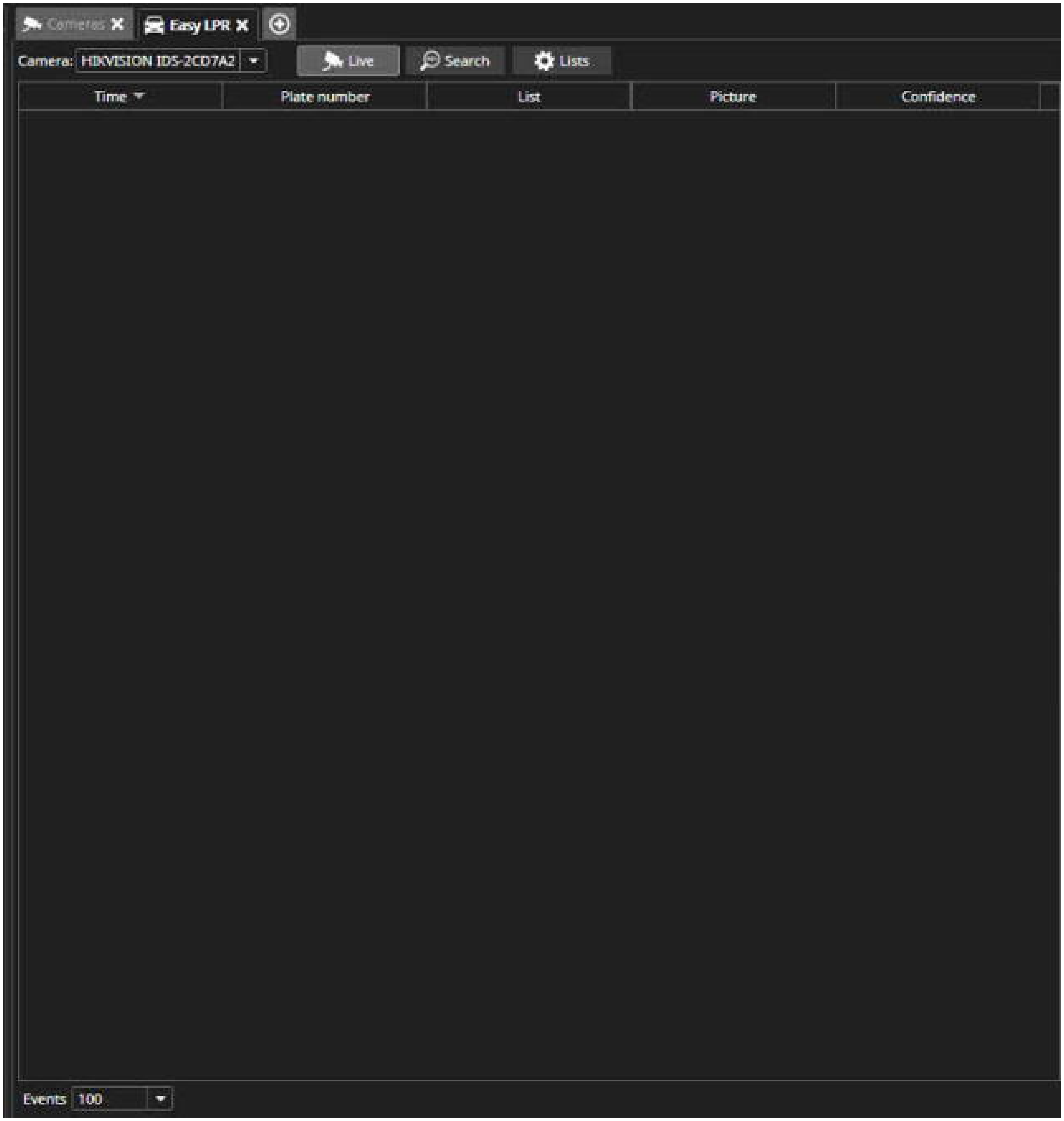
| Name | Priority | Trigger | Actions |
|---|---|---|---|
| Black list plate detected | Normal | Metadata on channel AXIS P1455-LE | Record video from Camera 1 |

Name: Black list plate detected
Description:
Priority: Normal
Requires Acknowledgment: No
Viewable in Profiles: v9.4
Trigger: Metadata on channel AXIS P1455-LE
Activate on metadata event Easy LPR: Black list plate detected
Actions: Record video from Camera 1
Resolution: 3840x2160
Recording rate: 25/s
Pre-event recording: Off
Post-event recording: On
Pre-event recording time: 0 s
Post-event recording time: 10 s
Calendar: The alarm is always enabled
Special Days:

## 4.6 Using Easy LPR

Easy LPR contains the following functionalities:

- Live monitoring from the 1 camera at the same time
- The search of the number plates
- Lists Management
- Digital output controlling based on lists



Camera: HIKVISION IDS-2CD7A2

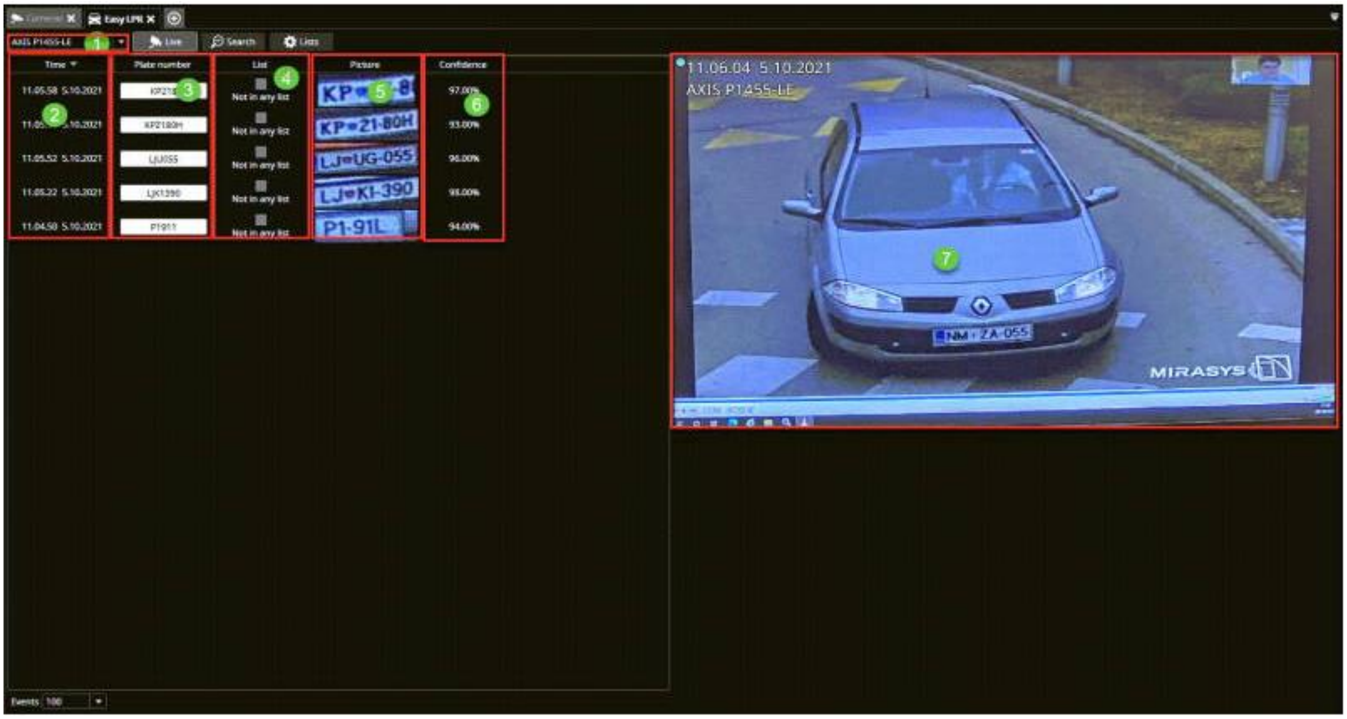
| Time | Plate number | List | Picture | Confidence |
|---|---|---|---|---|

Events 100

## 4.7 Live

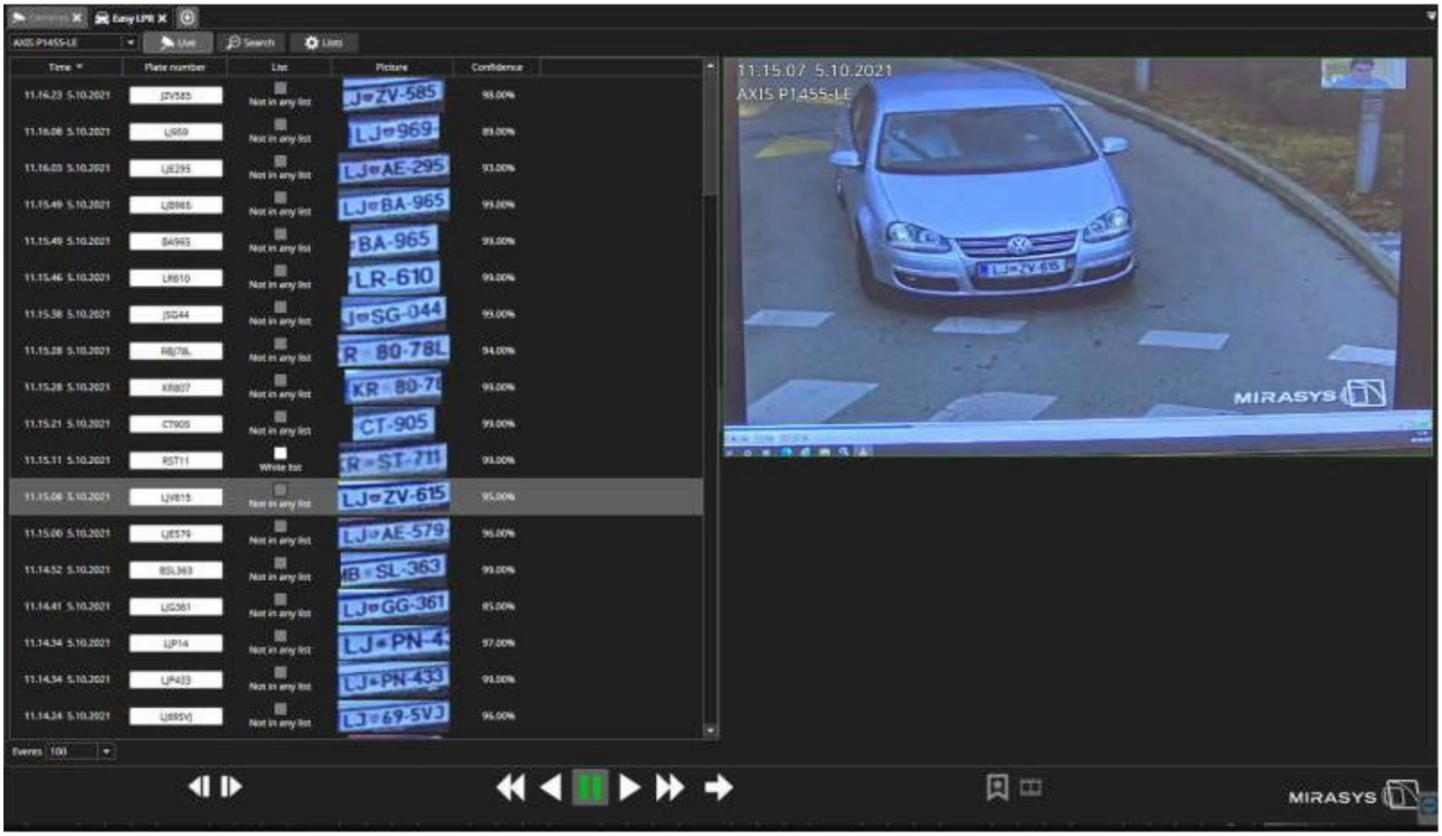
The live tab shows the following information:

1. The selection of the LPR camera
2. Time of the plate detection
3. Plate number
4. Plate list
5. Picture of the plate number
6. Confidence of the plate reading
7. Live view from the LPR camera



| Time | Plate number | List | Picture | Confidence |
|---|---|---|---|---|
| 11.05.58 5.10.2021 | KP21[illegible] | Not in any list | | 97.00% |
| 11.05.[illegible] 5.10.2021 | KP2180H | Not in any list | | 93.00% |
| 11.05.52 5.10.2021 | LJUG055 | Not in any list | | 96.00% |
| 11.05.22 5.10.2021 | LJKI390 | Not in any list | | 95.00% |
| 11.04.50 5.10.2021 | P1911 | Not in any list | | 94.00% |

When the plate information is clicked by the mouse, then the view changes to the playback mode and show the recorded situation.



| Time | Plate number | List | Picture | Confidence |
|---|---|---|---|---|
| 11.16.23 5.10.2021 | JZV585 | Not in any list | | 98.00% |
| 11.16.08 5.10.2021 | LJ969 | Not in any list | | 89.00% |
| 11.16.03 5.10.2021 | LJE295 | Not in any list | | 93.00% |
| 11.15.49 5.10.2021 | LJB965 | Not in any list | | 99.00% |
| 11.15.49 5.10.2021 | BA965 | Not in any list | | 99.00% |
| 11.15.46 5.10.2021 | LR610 | Not in any list | | 99.00% |
| 11.15.38 5.10.2021 | JSG44 | Not in any list | | 99.00% |
| 11.15.28 5.10.2021 | RBJ78L | Not in any list | | 94.00% |
| 11.15.28 5.10.2021 | KR807 | Not in any list | | 99.00% |
| 11.15.21 5.10.2021 | CT905 | Not in any list | | 99.00% |
| 11.15.11 5.10.2021 | RST11 | White list | | 99.00% |
| 11.15.06 5.10.2021 | LJV615 | Not in any list | | 95.00% |
| 11.15.00 5.10.2021 | LJE579 | Not in any list | | 96.00% |
| 11.14.52 5.10.2021 | BSL363 | Not in any list | | 99.00% |
| 11.14.41 5.10.2021 | LJG361 | Not in any list | | 85.00% |
| 11.14.34 5.10.2021 | LJP14 | Not in any list | | 97.00% |
| 11.14.34 5.10.2021 | LJP433 | Not in any list | | 99.00% |
| 11.14.24 5.10.2021 | LJ69SVJ | Not in any list | | 96.00% |

Events: 100

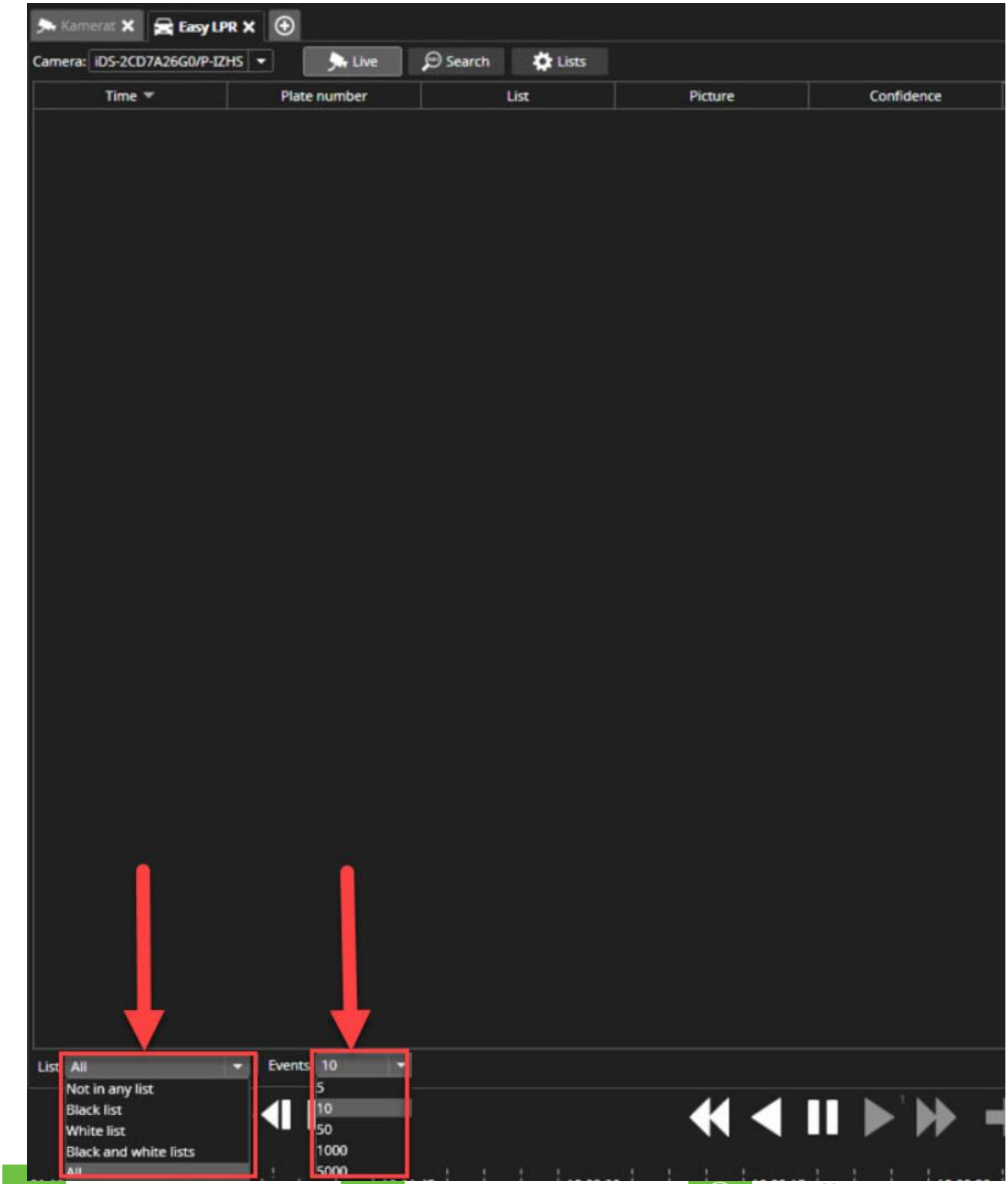
### 4.7.1 Filtering the Live view (supported since V9.5.0)

The user can which list are shown in the Live view. Options are:

- All
- Not in any list
- Black list
- White list
- Black list and White list

The user can set the amount of the result in the Live view. Options are:

- 5, 10, 50, 1000 and 5000



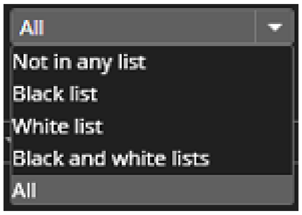
Camera: iDS-2CD7A26G0/P-IZHS

List: All

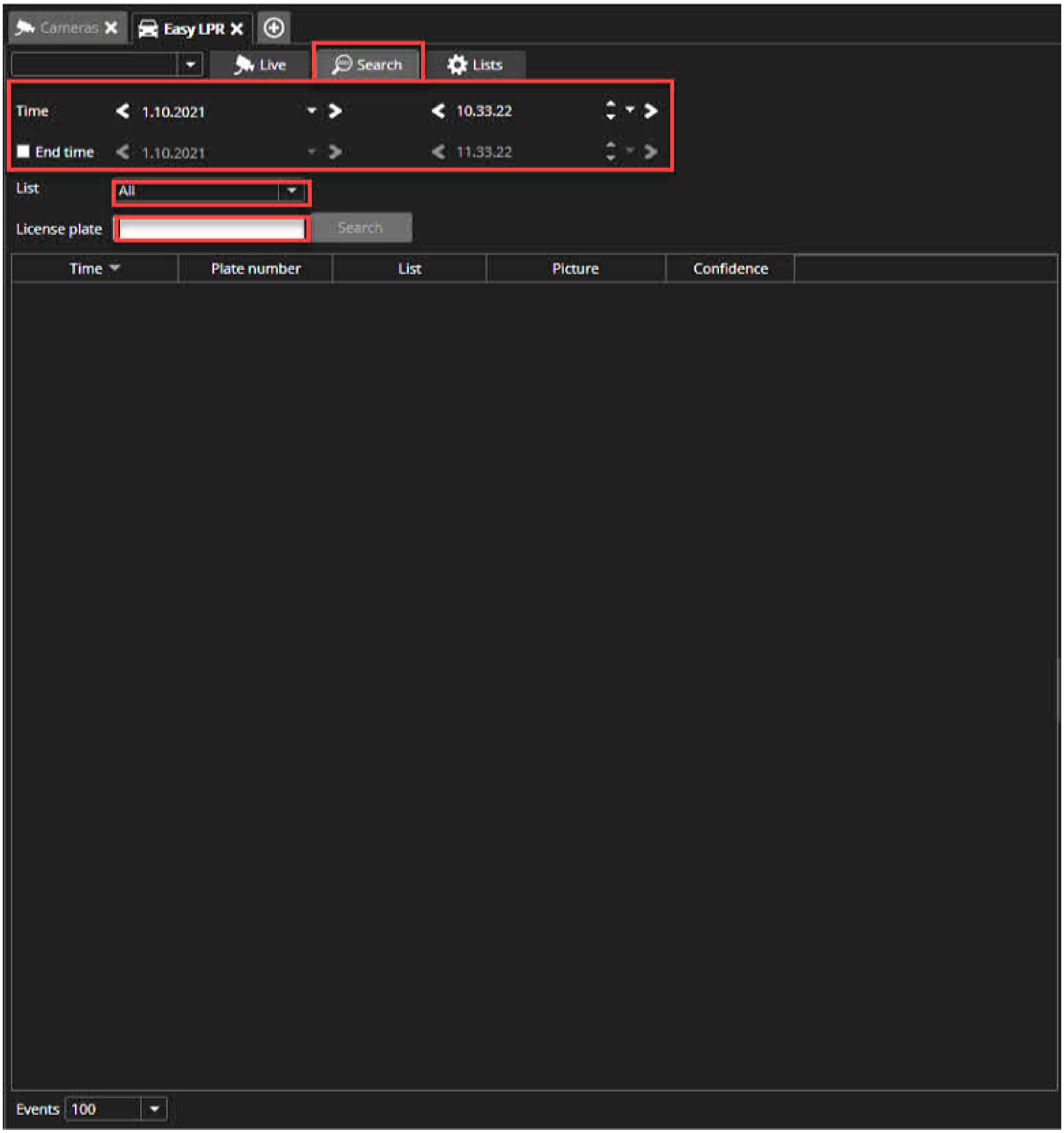
Events: 10

## 4.8 Searching License plates

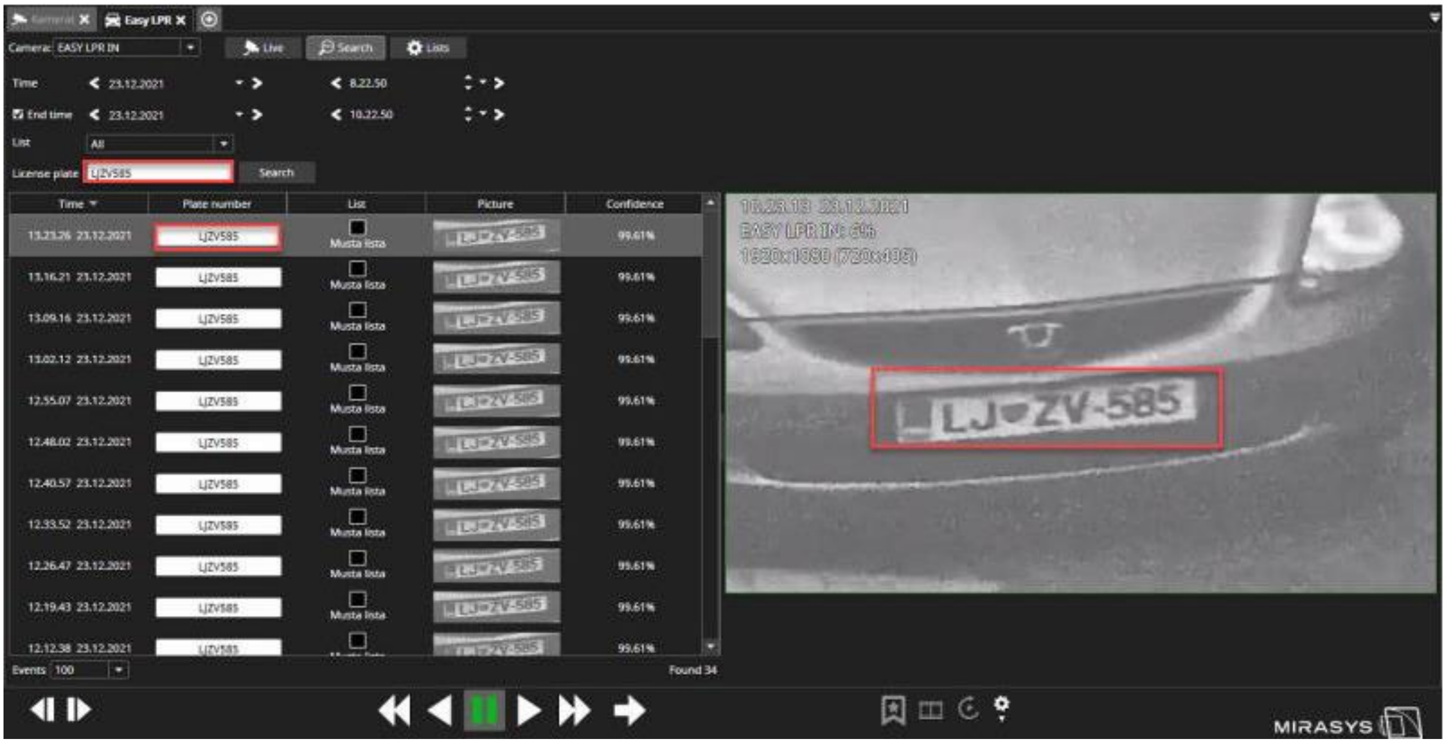
1. Open **Search** tab
2. Select LPR camera from the upper left corner
3. Select time and date
4. Enter **End time**, if needed
5. Select list for the search
    a. All
    b. Not in any list
    c. Black list
    d. White list
    e. Black and white list



1. Enter license plate(partial information is also accepted)
2. Click **Search**


Cameras
Easy LPR
Live
Search
Lists
Time
1.10.2021
10.33.22
End time
1.10.2021
11.33.22
List
All
License plate
Search
Time
Plate number
List
Picture
Confidence
Events
100

Search will show all results. The user can playback selected time and use all normal playback functions.



Camera: EASY LPR IN

Time 23.12.2021 8.22.50

☑ End time 23.12.2021 10.22.50

List: All

License plate: LJZV585 Search

| Time | Plate number | List | Picture | Confidence |
|---|---|---|---|---|
| 13.23.26 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 13.16.21 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 13.09.16 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 13.02.12 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.55.07 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.48.02 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.40.57 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.33.52 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.26.47 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.19.43 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |
| 12.12.38 23.12.2021 | LJZV585 | Musta lista | LJ ZV-585 | 99.61% |

Events 100 Found 34

## 4.9 Lists

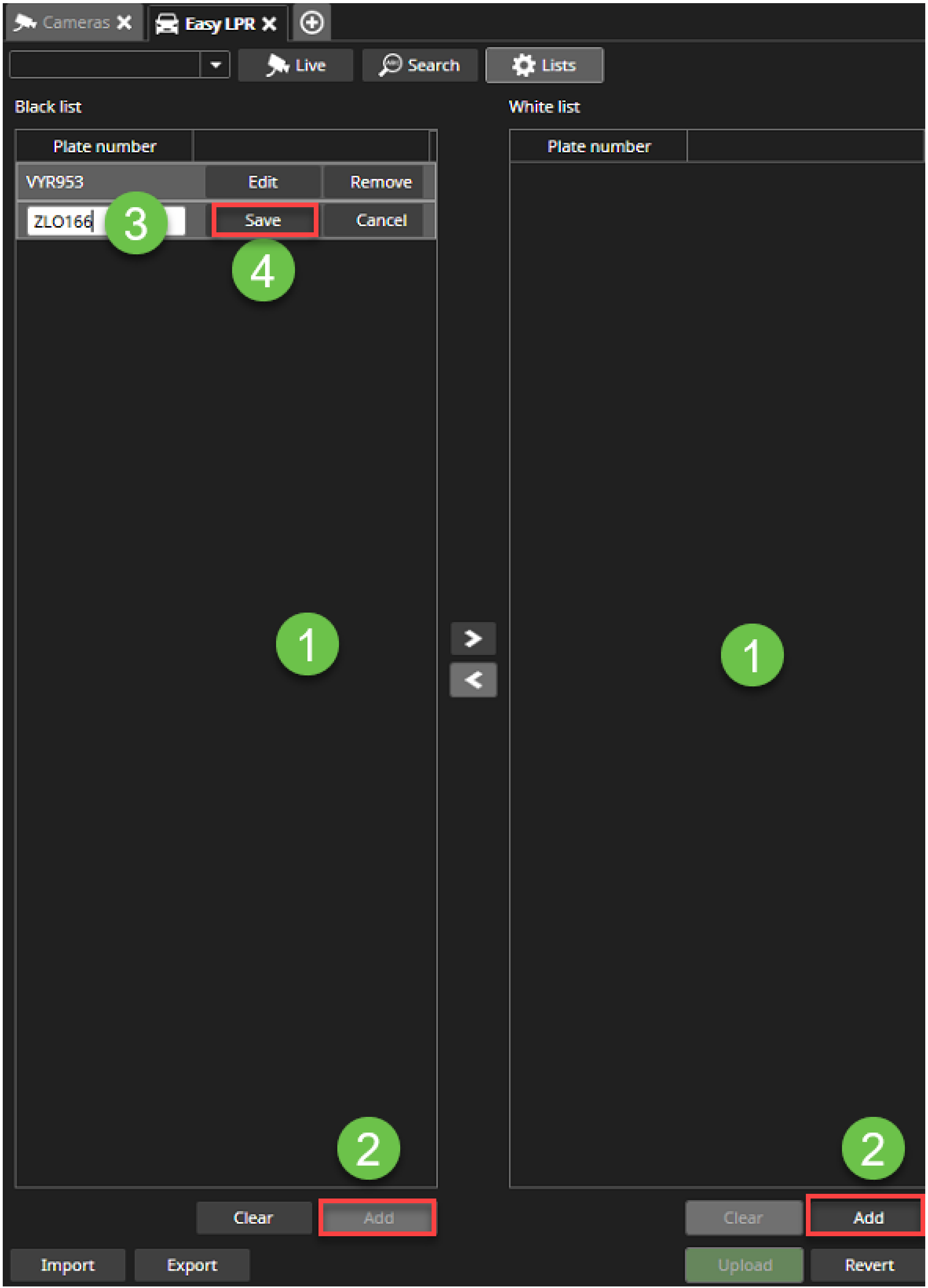
### 4.9.1 With the Easy LPR Lists Management, the users can do the following actions:

- Add plate number
- Edit plate numbers
- Move plate numbers between the lists
- Export plate numbers from the Spotter to the PC(**CSV**)
- Import edited plate number lists to the Spotter
- Upload lists from the Spotter to the LPR cameras

### 4.9.2 Please remember to upload lists to the cameras after any change.

### 4.9.3 Adding Plate number

1. Select the **Black list** or **White list**
2. Click **Add**
3. Type the plate number
4. Click **Save**

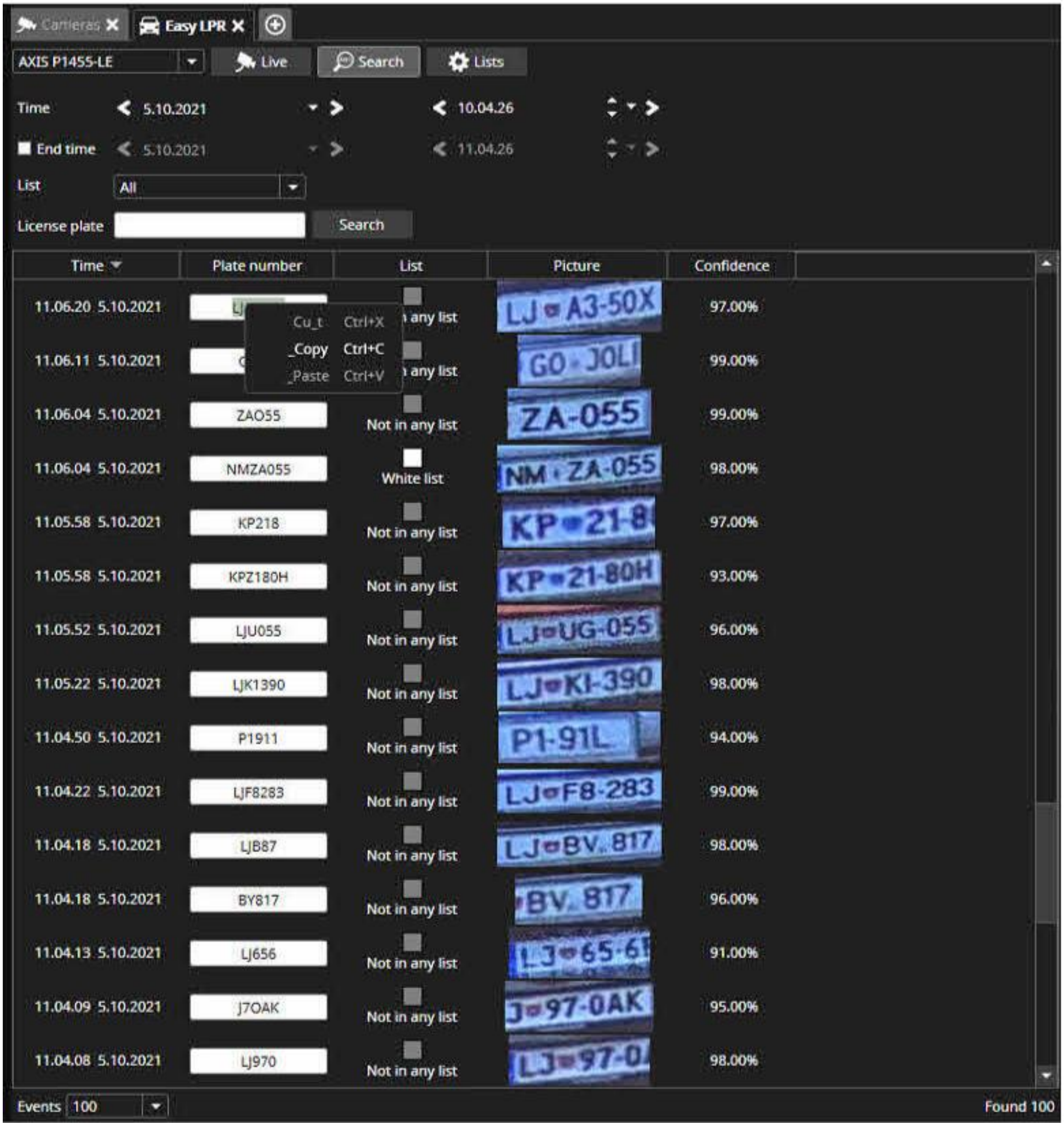


VYR953

ZLO166

#### *4.9.3.1 Adding plate number from the search view*

1. Double-click plate number field
2. Right mouse click top of the plate number
3. Click **Copy**



Time 5.10.2021 10.04.26

□ End time 5.10.2021 11.04.26

List All

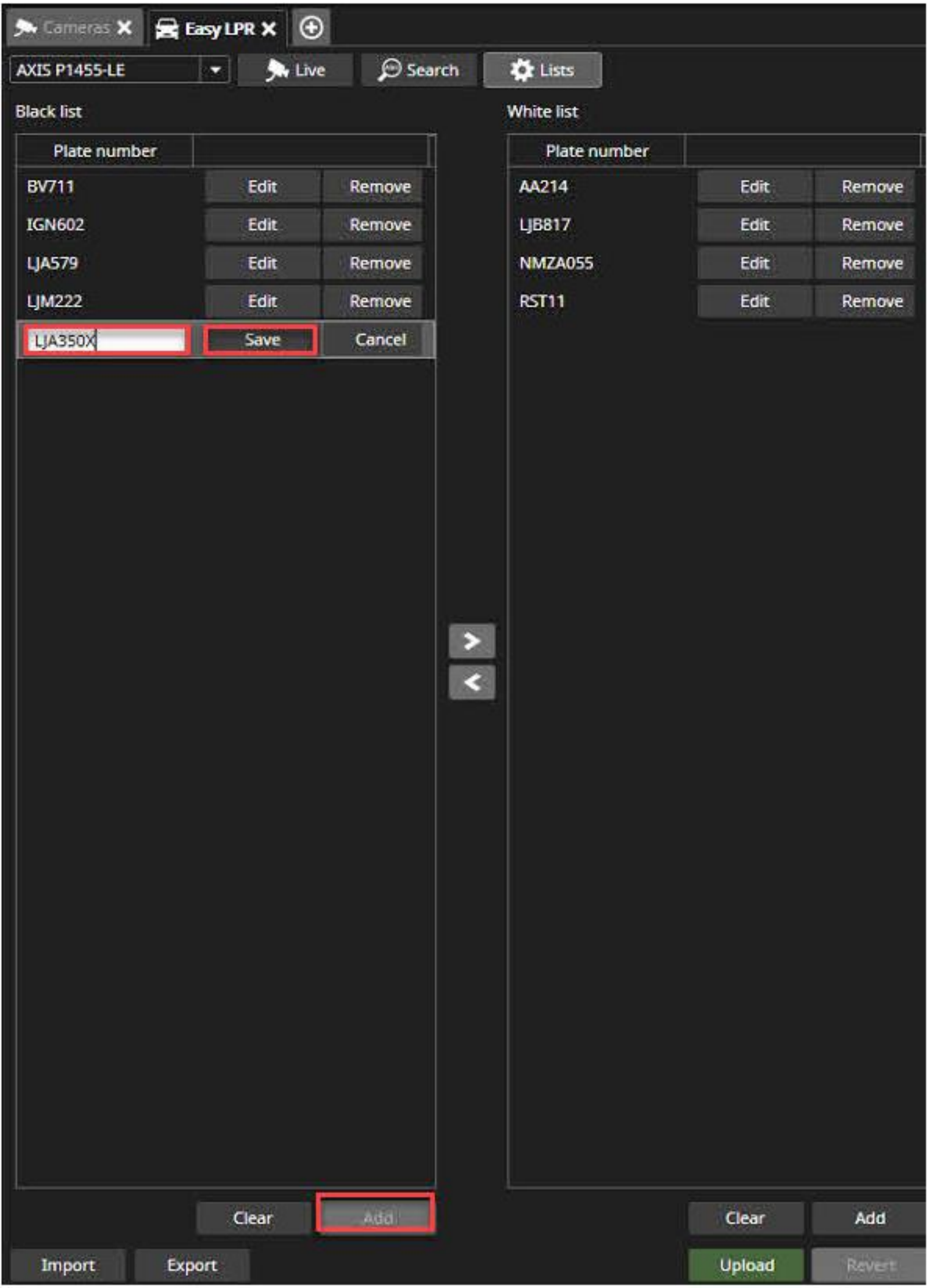
License plate Search

| Time | Plate number | List | Picture | Confidence |
|---|---|---|---|---|
| 11.06.20 5.10.2021 | LJ | any list | | 97.00% |
| 11.06.11 5.10.2021 | G | any list | | 99.00% |
| 11.06.04 5.10.2021 | ZAO55 | Not in any list | | 99.00% |
| 11.06.04 5.10.2021 | NMZA055 | White list | | 98.00% |
| 11.05.58 5.10.2021 | KP218 | Not in any list | | 97.00% |
| 11.05.58 5.10.2021 | KPZ180H | Not in any list | | 93.00% |
| 11.05.52 5.10.2021 | LJU055 | Not in any list | | 96.00% |
| 11.05.22 5.10.2021 | LJK1390 | Not in any list | | 98.00% |
| 11.04.50 5.10.2021 | P1911 | Not in any list | | 94.00% |
| 11.04.22 5.10.2021 | LJF8283 | Not in any list | | 99.00% |
| 11.04.18 5.10.2021 | LJB87 | Not in any list | | 98.00% |
| 11.04.18 5.10.2021 | BY817 | Not in any list | | 96.00% |
| 11.04.13 5.10.2021 | LJ656 | Not in any list | | 91.00% |
| 11.04.09 5.10.2021 | J7OAK | Not in any list | | 95.00% |
| 11.04.08 5.10.2021 | LJ970 | Not in any list | | 98.00% |

Events 100

Found 100

4. Open **Lists**
5. Select current list
6. Click **Add**
7. Paste plate number
8. Click **Save**



Black list

| Plate number |
|---|
| BV711 |
| IGN602 |
| LJA579 |
| LJM222 |
| LJA350X |

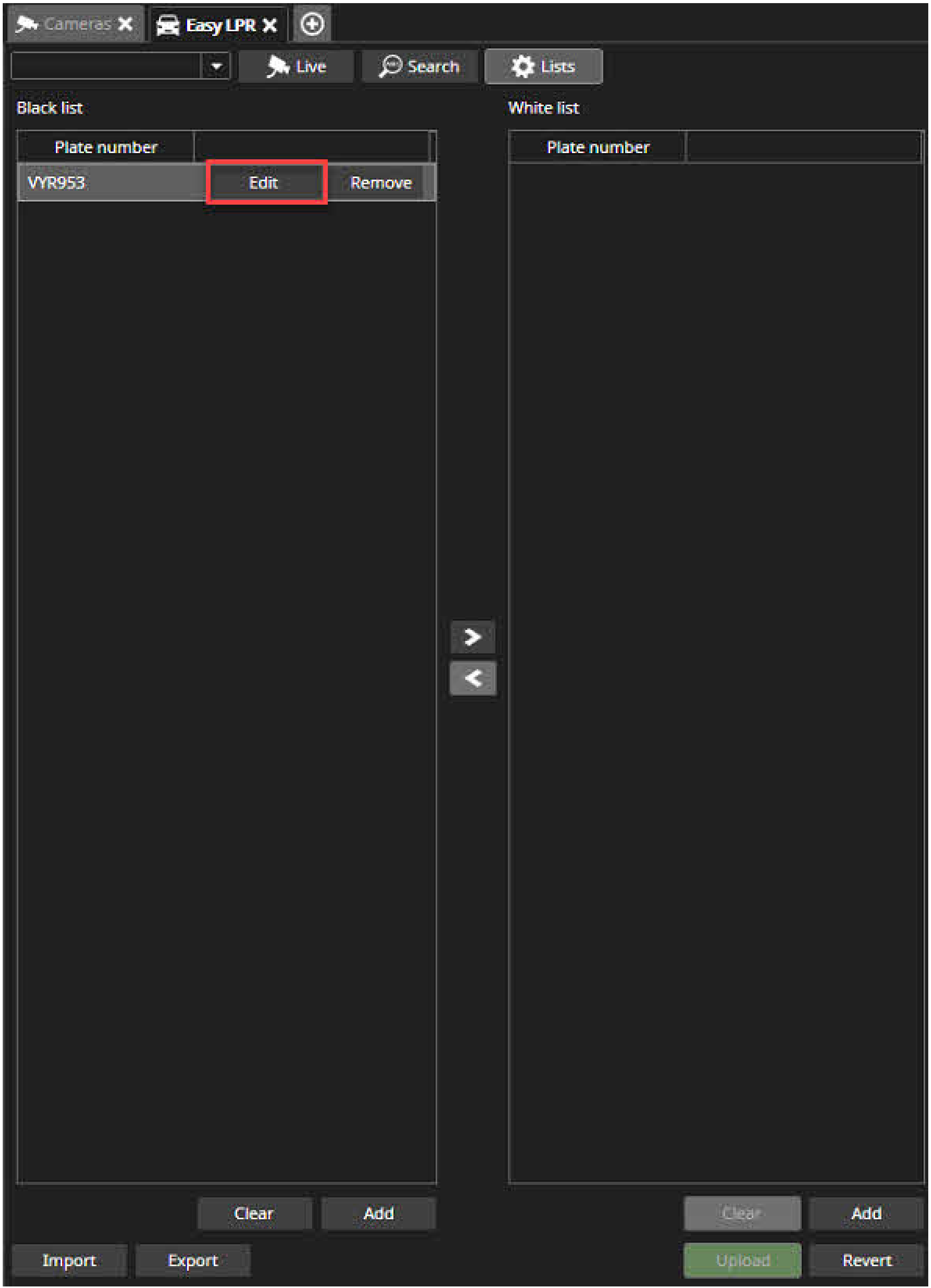
White list

| Plate number |
|---|
| AA214 |
| LJB817 |
| NMZA055 |
| RST11 |

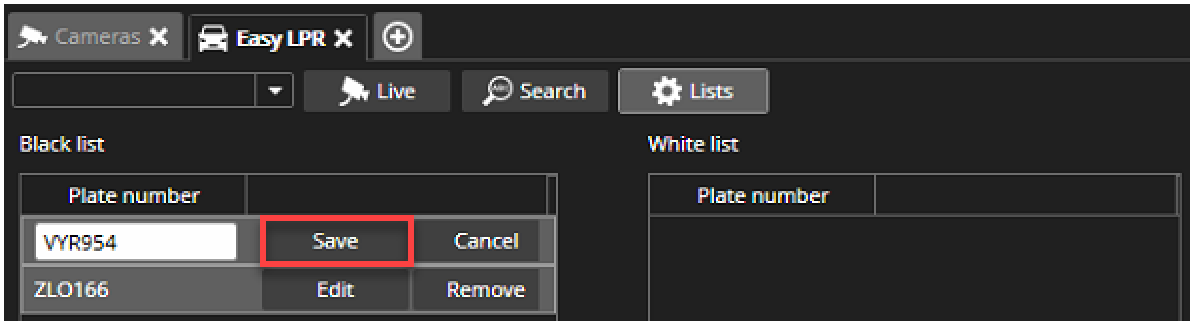
### 4.9.4 Editing Plate Number

1. Select the plate number

2. Click **Edit**

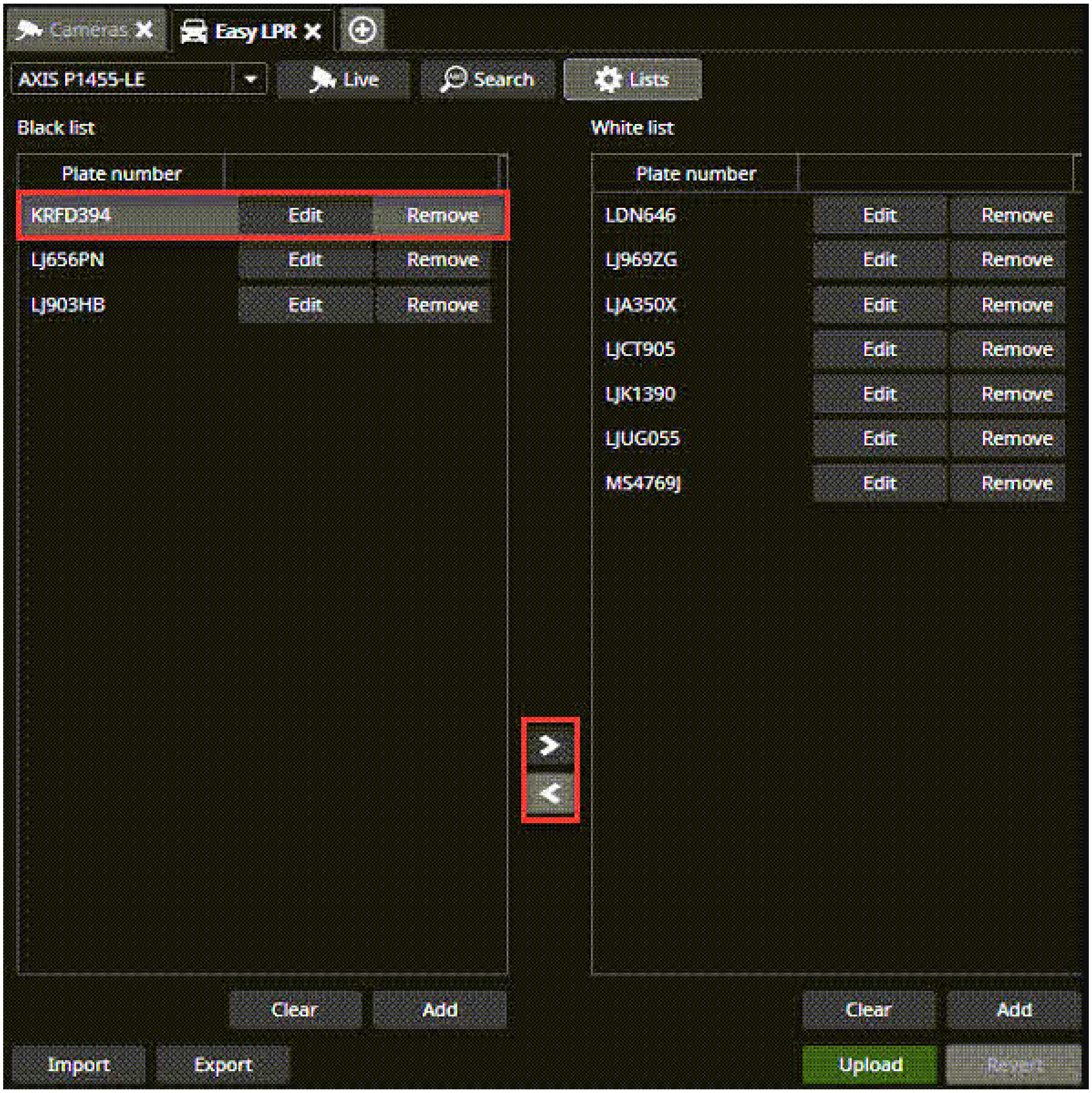

Cameras
Easy LPR
Live
Search
Lists
Black list
White list
Plate number
VYR953
Edit
Remove
Plate number
>
<
Clear
Add
Clear
Add
Import
Export
Upload
Revert

3. Do the modification and click **Save**



### 4.9.5 Moving Plate Number between the lists

1. Select the plate number from the list
2. Click arrow to move needed list



Black list

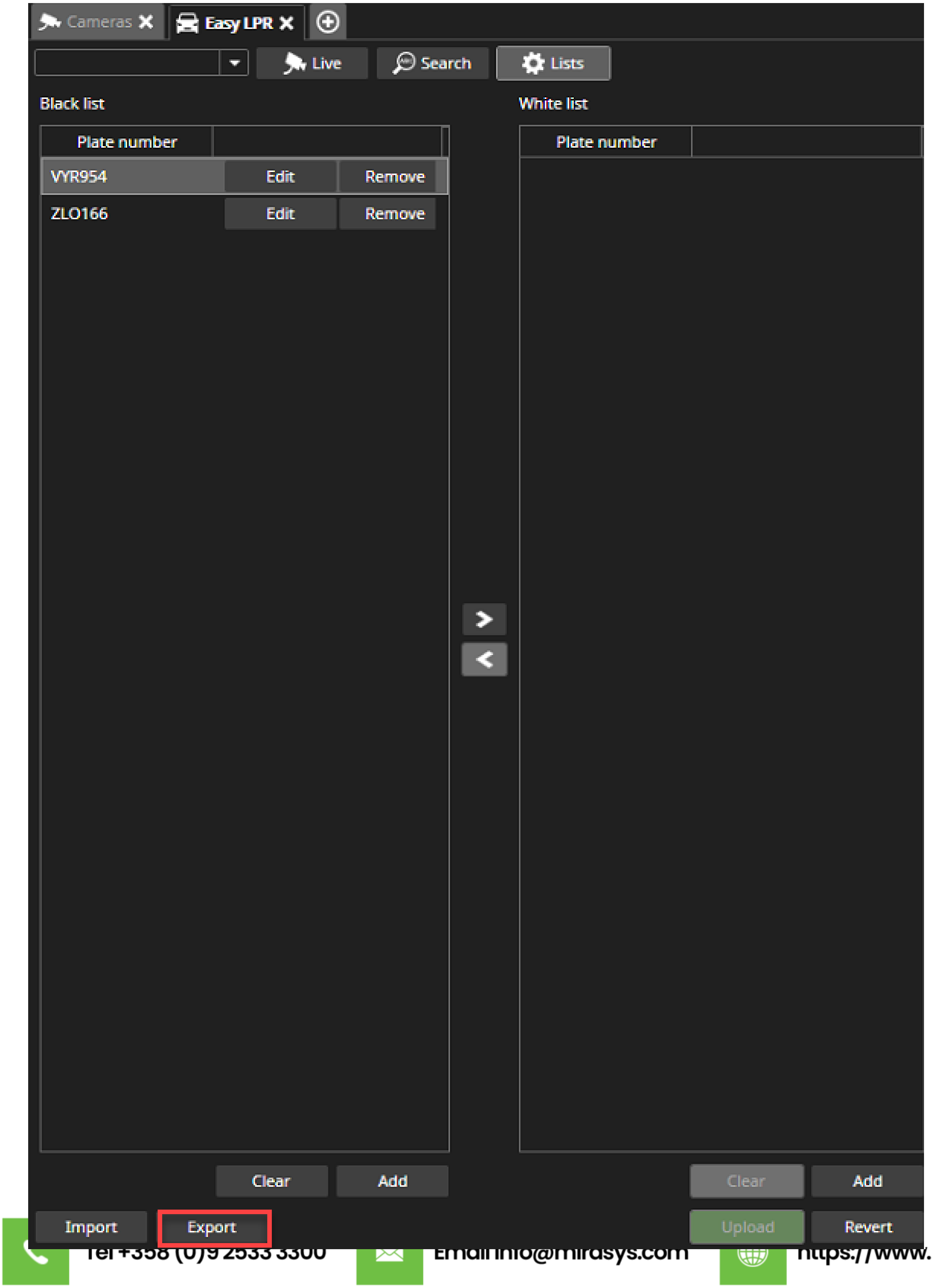
| Plate number | | |
|---|---|---|
| KRFD394 | Edit | Remove |
| LJ656PN | Edit | Remove |
| LJ903HB | Edit | Remove |

White list

| Plate number | | |
|---|---|---|
| LDN646 | Edit | Remove |
| LJ969ZG | Edit | Remove |
| LJA350X | Edit | Remove |
| LJCT905 | Edit | Remove |
| LJK1390 | Edit | Remove |
| LJUG055 | Edit | Remove |
| MS4769J | Edit | Remove |

### 4.9.6 Export Plate Number lists

1. Click **Export**



Black list

| Plate number | | |
|---|---|---|
| VYR954 | Edit | Remove |
| ZLO166 | Edit | Remove |

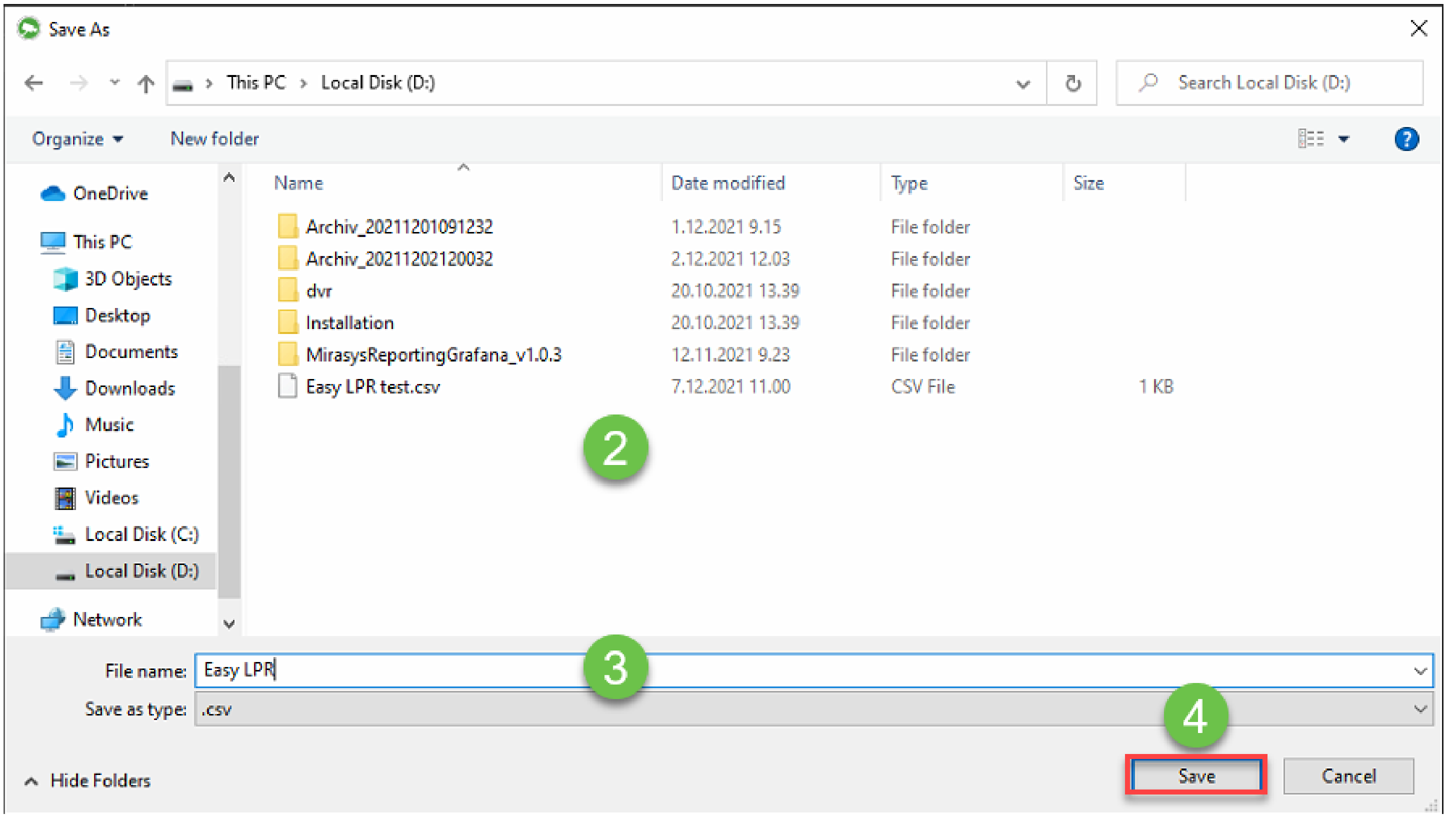
White list

| Plate number | |
|---|---|

2. Define the destination folder

3. Set the file name(.csv)

4. Click **Save**



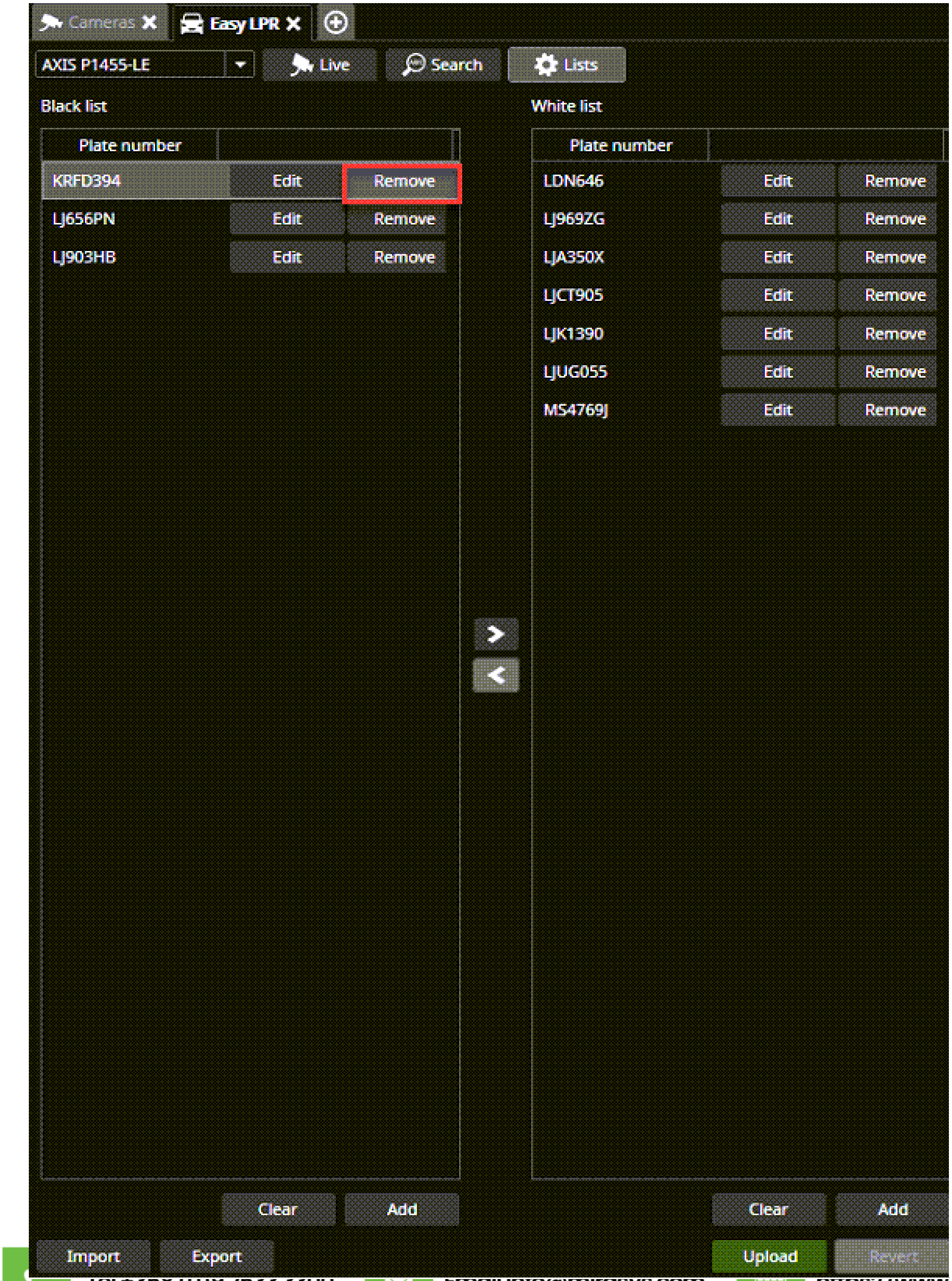
| Name | Date modified | Type | Size |
|---|---|---|---|
| Archiv_20211201091232 | 1.12.2021 9.15 | File folder | |
| Archiv_20211202120032 | 2.12.2021 12.03 | File folder | |
| dvr | 20.10.2021 13.39 | File folder | |
| Installation | 20.10.2021 13.39 | File folder | |
| MirasysReportingGrafana_v1.0.3 | 12.11.2021 9.23 | File folder | |
| Easy LPR test.csv | 7.12.2021 11.00 | CSV File | 1 KB |

File name: Easy LPR

Save as type: .csv

### 4.9.7 Removing Plate Numbers

1. Select the plate number from the list
2. Click Remove



Black list

| Plate number | | |
|---|---|---|
| KRFD394 | Edit | Remove |
| LJ656PN | Edit | Remove |
| LJ903HB | Edit | Remove |

White list

| Plate number | | |
|---|---|---|
| LDN646 | Edit | Remove |
| LJ969ZG | Edit | Remove |
| LJA350X | Edit | Remove |
| LJCT905 | Edit | Remove |
| LJK1390 | Edit | Remove |
| LJUG055 | Edit | Remove |
| MS4769J | Edit | Remove |

### 4.9.8 With the import, the user can import a large number of plate numbers at the same time

1. Open exported CSV file

CSV content is shown below:

Plate number, List (1 = black list / 2 = white list)

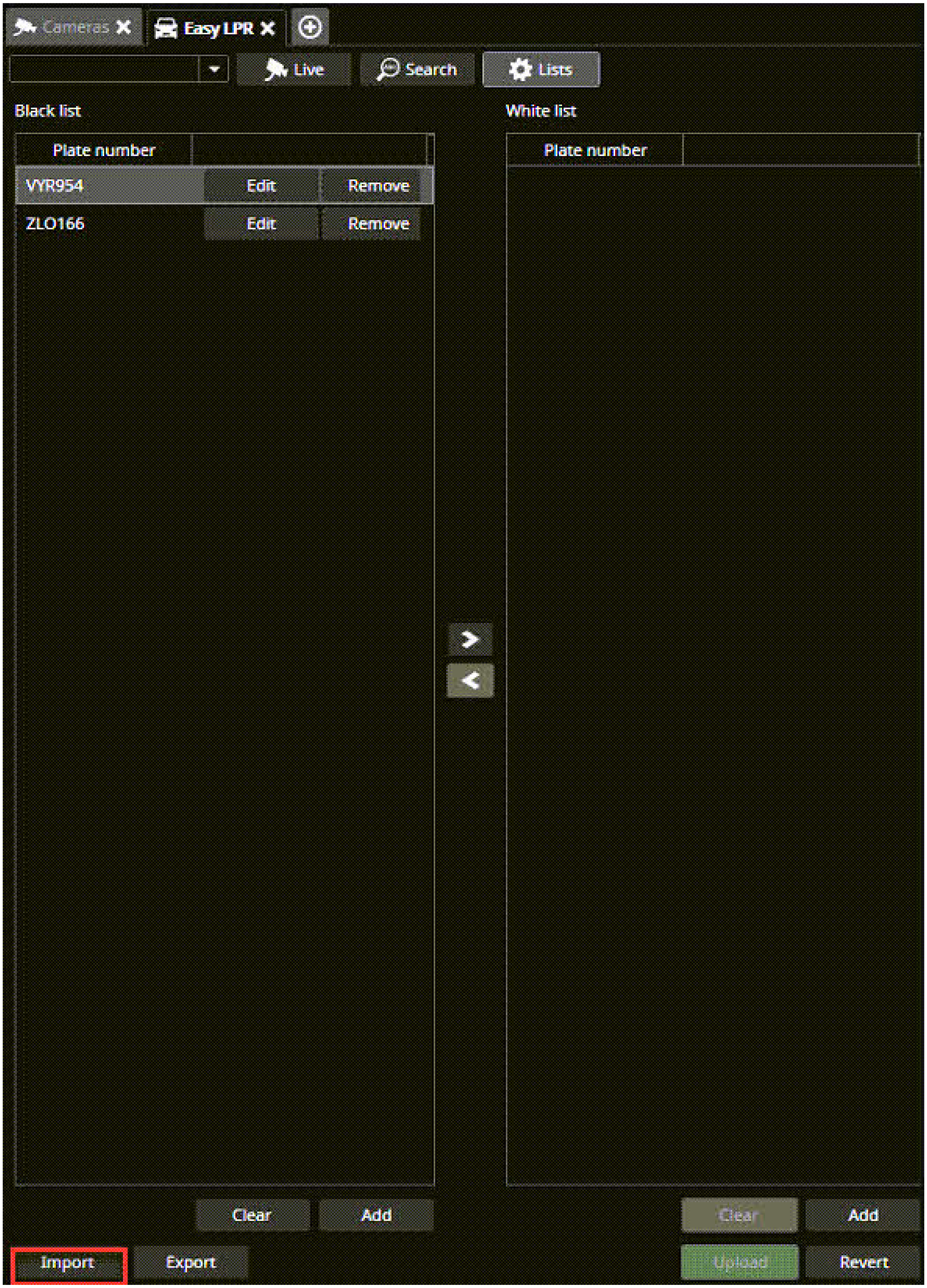
LJ656PN,1

LJ731CV,1

LJZV585,1

LJZV584,2

1. Add a new line with format ZLO166,2  for each new plate number
2. Select correct list(**List 1 = Black list, List 2 = White Lis**t)
3. Save changes
4. Click **Import**


Cameras
Easy LPR
Live
Search
Lists
Black list
Plate number
VYR954
Edit
Remove
ZLO166
Edit
Remove
White list
Plate number
Clear
Add
Clear
Add
Import
Export
Upload
Revert

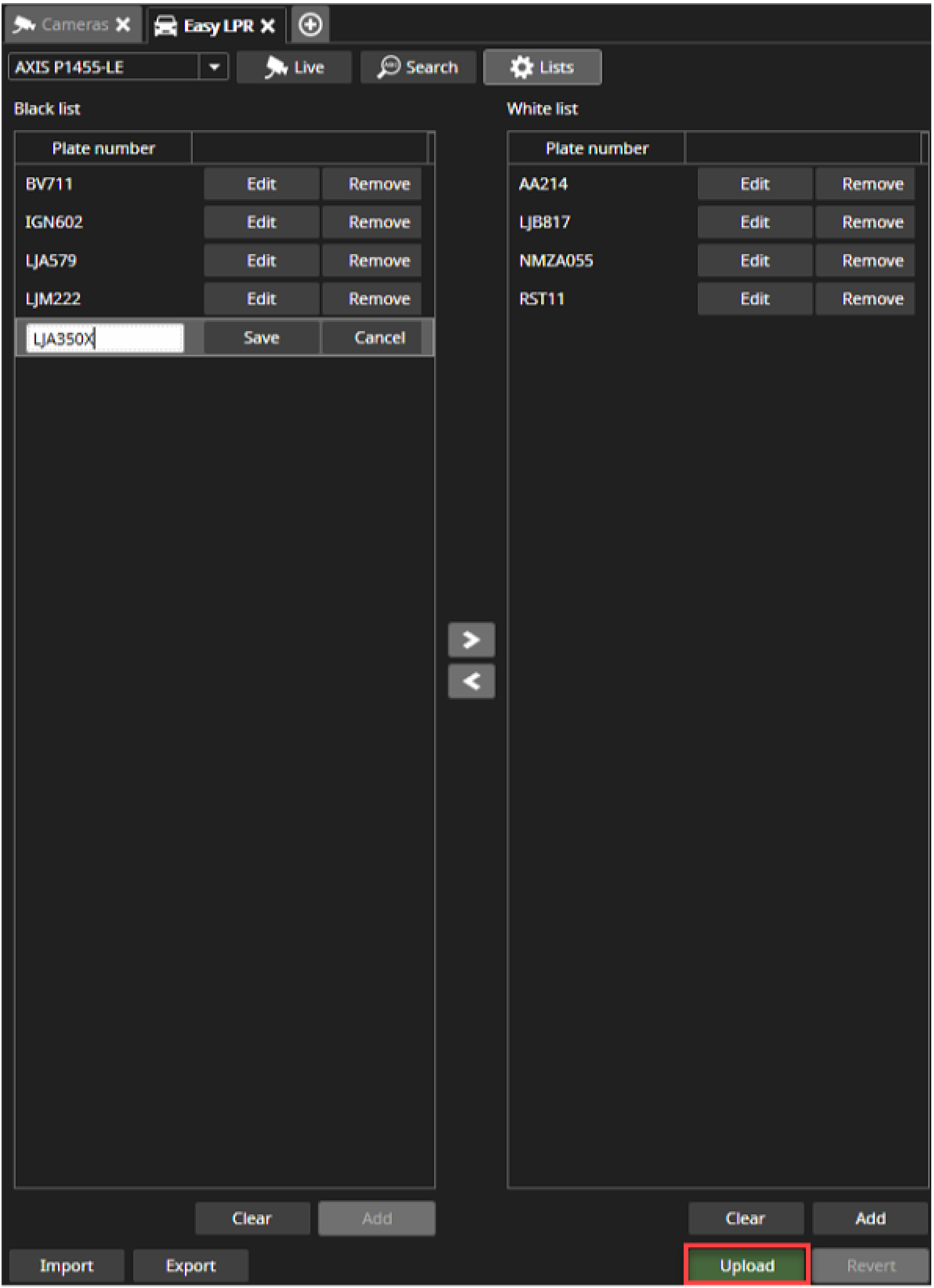
5. Browse to the location of the CSV file

6. Select the file and click **Open**

### 4.9.9 With the upload, the user can upload created black & white lists to the camera

1. Select camera, which list will be uploaded

2. Click **Upload**



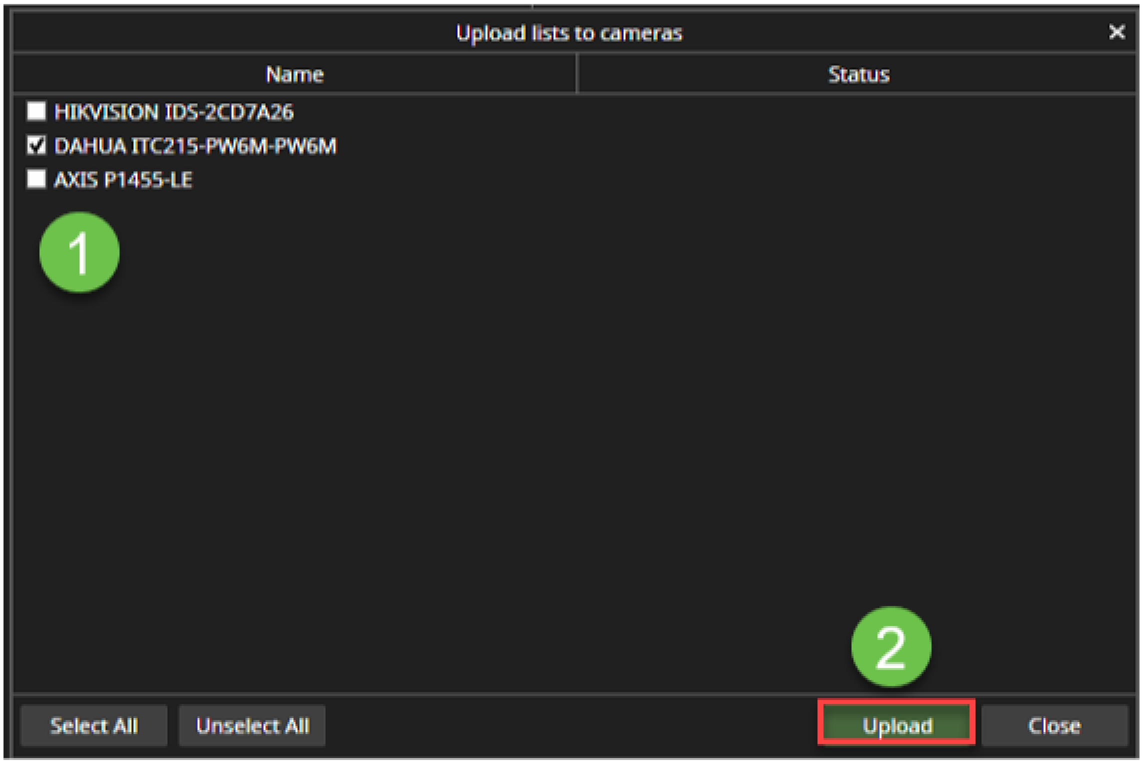
Black list

| Plate number | | |
|---|---|---|
| BV711 | Edit | Remove |
| IGN602 | Edit | Remove |
| LJA579 | Edit | Remove |
| LJM222 | Edit | Remove |
| LJA350X | Save | Cancel |

White list

| Plate number | | |
|---|---|---|
| AA214 | Edit | Remove |
| LJB817 | Edit | Remove |
| NMZA055 | Edit | Remove |
| RST11 | Edit | Remove |

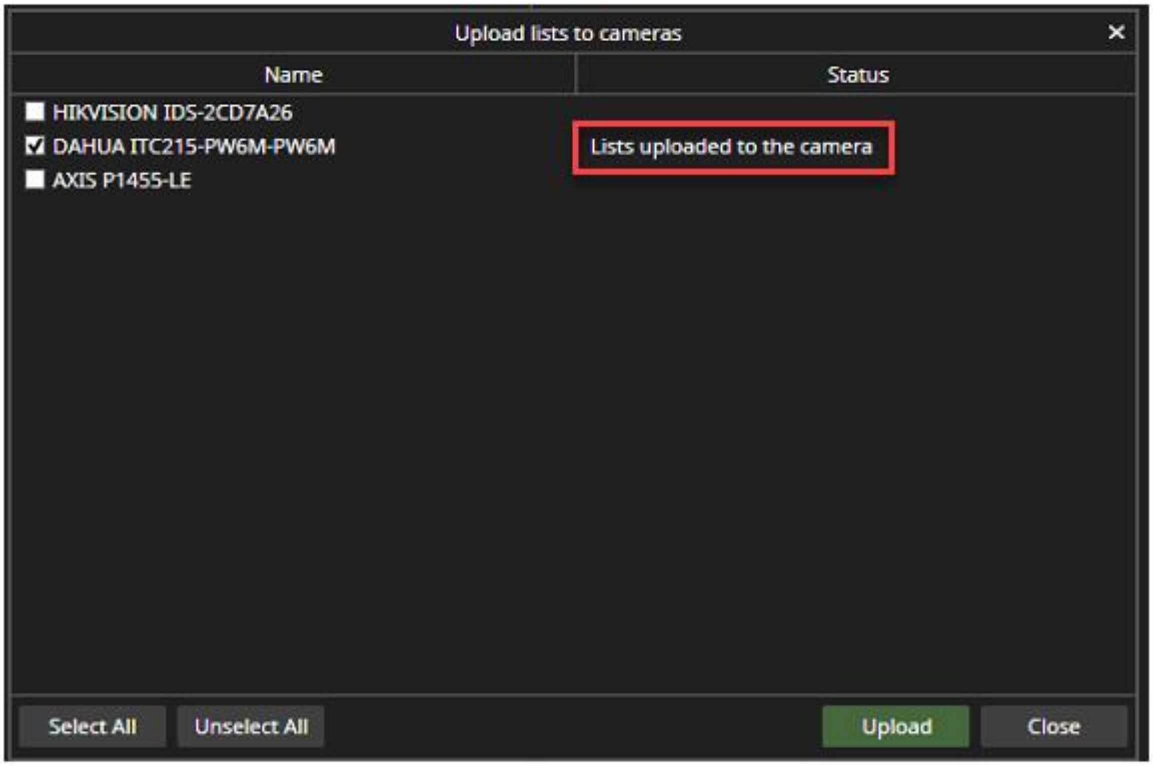
1. Select cameras, where lists are uploaded
2. Click **Upload**



Upload lists to cameras

| Name | Status |
|---|---|
| □ HIKVISION IDS-2CD7A26 | |
| ☑ DAHUA ITC215-PW6M-PW6M | |
| □ AXIS P1455-LE | |

After the upload, the status field shows information **List uploaded to the camera**

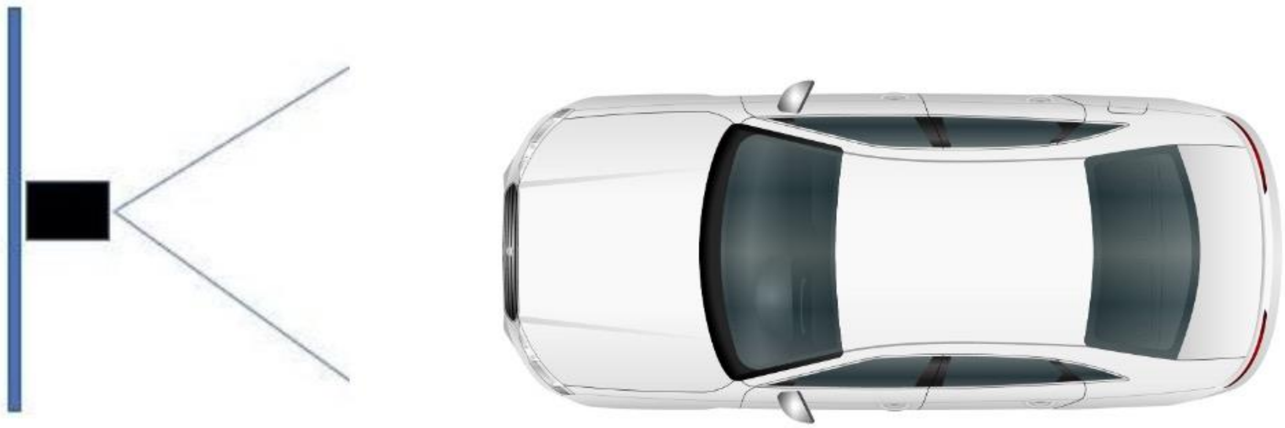


Upload lists to cameras

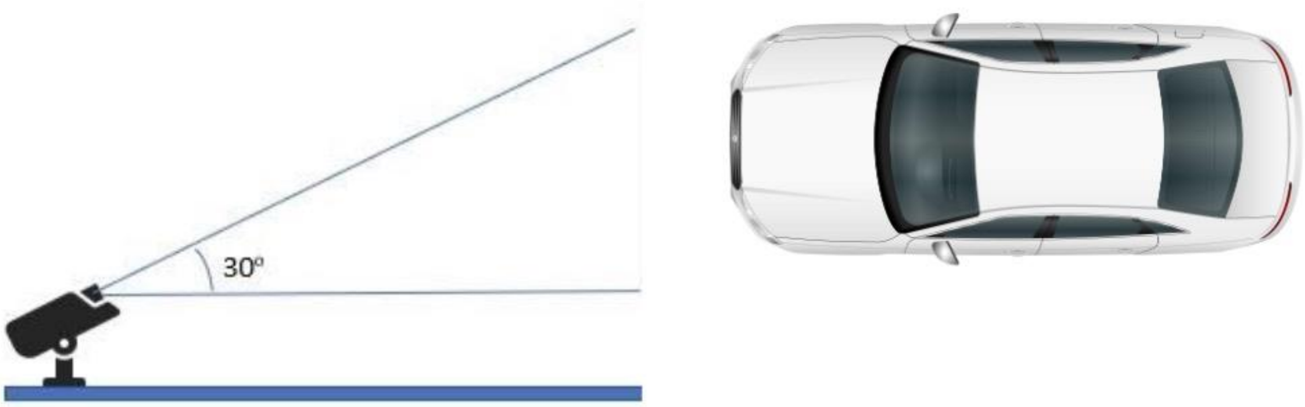
| Name | Status |
|---|---|
| □ HIKVISION IDS-2CD7A26 | |
| ☑ DAHUA ITC215-PW6M-PW6M | Lists uploaded to the camera |
| □ AXIS P1455-LE | |

# 5 LPR Camera Installation tips

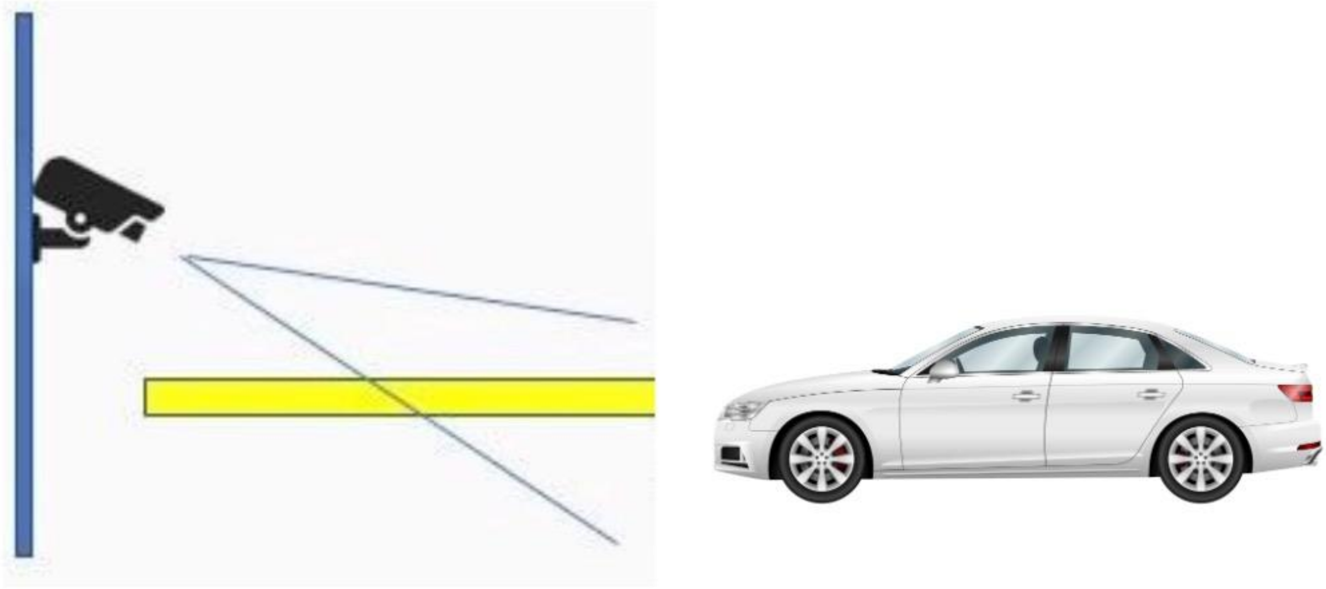
## 5.1 It is recommended to install the camera in the center of the vehicle



## 5.2 IF THE CAMERA IS INSTALLED ON THE SIDE OF THE ROAD OR LANE, THE ANGLE SHOULD NOT EXCEED 30 DEGREES.



## 5.3 The camera should be installed higher than the vehicle headlights so that the vehicle's headlights don't point directly at the camera



## 5.4 Ensure the license plate width is at least 120 pixels and height at least 50 pixels

Height at least 50 pixels

Width at least 120 pixels

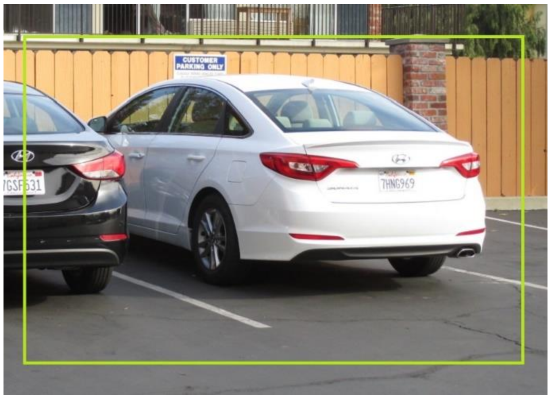
## 5.5 License plate tilt angle must be within +/- 10 degrees

## 5.6 LPR settings in the System Manager application

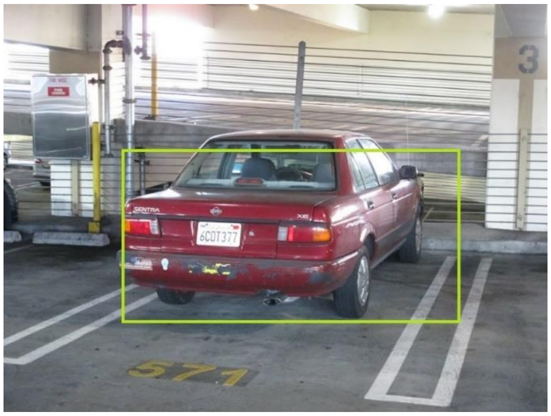
Ensure the correct region (Americas / Eurasia) is selected.

### 5.6.1 Setting the region of interest

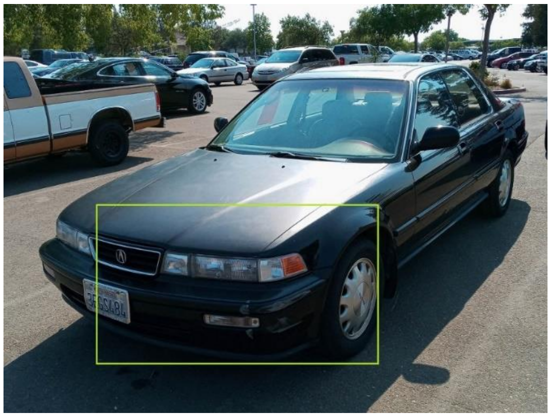
Region of interest is used to define where detection will find license plates.



Leave some margin to the region of interest to not detect partially visible plates.



The whole license plate is inside the region of interest, and the plate is detected.



The license plate is not completely inside the region of interest, and the plate is not detected.

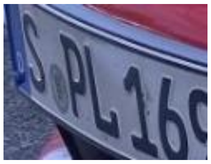
### 5.6.2 Enabling country recognition

In many countries letter **O** is similar to the number **0**, and the letter **I** looks the same as the number **1**. Enabling country detection improves detection accuracy in these cases.

For example, the format for Brazil plate number is “abc1d23”.

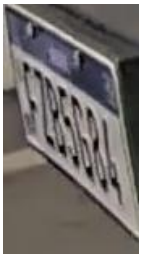
## 5.7 Common problems and solutions

### 5.7.1 Incomplete license plate



Solution: Don’t set the region of interest too close to image borders.

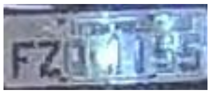
### 5.7.2 View angle makes plate numbers unreadable



Solution 1: Move the camera to a better place.

Solution 2: Set the region of interest so that the plate is detected when it is visible from better angle.

### 5.7.3 Other vehicles headlights reflect from license plate



Solution 1: Move the camera to a better place.

Solution 2: Set the region of interest so that the plate is detected when other vehicles’ headlights don’t point to the license plate direction.

### 5.7.4 The license plate is too small

Solution 1: Move the camera to a better place or zoom in.

Solution 2: Set the region of interest so that the plate is detected when the vehicle is nearer to the camera.

Solution 3: Increase the minimum plate height value in LPR settings so that small plate detections are ignored.

### 5.7.5 The license plate is blurry

Solution 1: Adjust sharpness and increase shutter speed in the camera's settings.

Solution 2: Increase the lighting of the area.

### 5.7.6 The license plate is overexposed

Solution 1: Adjust camera's image settings.

Solution 2: Check the camera installation location and move it higher so that headlights doesn't reflect from the plate.

# 6 Mirasys VCA Guide

## 6.1 About Mirasys VCA

Mirasys VCA (Video Content Analytics) comprises a set of real-time video analytics solutions that utilizes advanced image processing algorithms to turn video into actionable intelligence.
The product's core is an advanced object recognition and tracking engine that continually tracks moving and stationary targets.

The tracking engine features built-in robustness to environmental nuisance conditions such as changing illumination, moving foliage, rippling water, etc.

Mirasys VCA is a generic name for a suite of video analytics add-on product options that include functionality such as:

- Ability to detect aggressive behavior.

- Ability to detect falls.
- Ability to detect directional crossing.
- Ability to detect repeated behavior.
- Ability to detect persons with their hands up.
- Deep learning object tracker classes are now person, bus, motorcycle, bicycle, car, van, truck, forklift, and bag for triggering alarms
- Support for different event state possibilities: start (default), on, stop. Note that the states on and stop are only available if defined in VCA Core config files.

Other Features are:

### 6.1.1 Motion object tracking

Motion-data based object highlighting and tracking, auto-zoom functionality.
The motion data is produced by server-based, hermeneutic motion detection.

### 6.1.2 Tripwire counting

In addition to motion object tracking functionality, line counting for over-head installed cameras, and Spotter client-based counter visualization.

### 6.1.3 Object behaviour/attributes detection

In addition to the functionality mentioned above, to continuously track and classify moving and stationary targets and features a full suite of rule-based filters including as enter, exit, appear, disappear, stopped objects, directionality constraints, object counting, loitering, object type and object speed.

Multiple filters and rules are supported on any combination of multiple overlapping detection zones, in addition to an advanced people tracking engine optimized for tracking people in cluttered indoor scenes such as retail scenarios. Includes specific high accuracy counting functions optimized for use in busy scenes.

### 6.1.4 Related analytics options

Available as separate applications, products or through project-based integrations:

- Camera-based (built-in, edge) analytics support selected camera manufacturers and their functionality through manufacturer-specific integration connectors.
- Audio analytics technologies refer to software for extracting information and meaning from audio signals, such as detecting sounds of breaking glass, etc.
- Facial recognition technologies refer to software or camera feature for automatically identifying or verifying the identity, age, gender, etc., of a person from video footage.
- Number plate recognition technologies (ANPR/LPR) refer to software or camera features for automatically identifying vehicle or container numbers.

## 6.2 Quick start guide for VCA

This user guide documents each topic in detail. However, to get started quickly, the essential topics are listed below.

### 6.2.1 The following steps should be executed for each server:

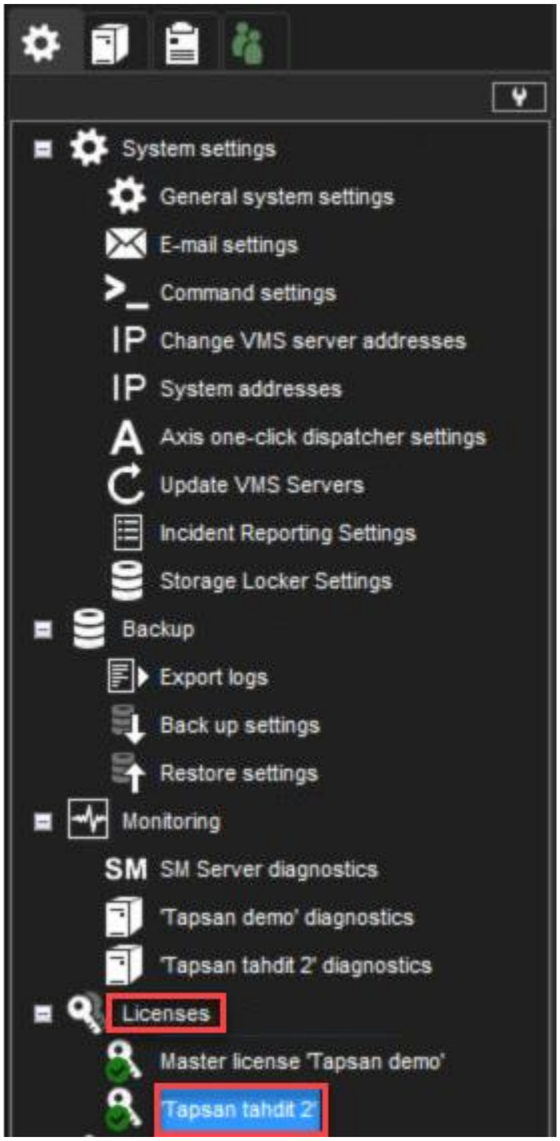
1. Decide upon the VCA functionality that meets your requirements. For guidance or consult your Mirasys representative or check the Mirasys VCA training.
2. Acquire and install a Mirasys VMS system and the related software license key with other required features enabled.
3. Add and configure the video cameras you intend to use for VCA and enable the VCA capability in the camera settings.
4. Enable hermeneutic motion detection mode for each camera used for VCA.
5. Export the VCA core HW GUID and obtain the VCA activation license code from Mirasys and activate Mirasys VCA with these licenses.
6. Calibrate each camera in VCA settings if object classification is required.
7. Configure the detection zone and rules for each camera.
8. If required configure alarms based on the VCA events.
9. Verify VCA functionality visualisation using the Spotter for Windows application.

## 6.3 Prerequisites for Mirasys VCA

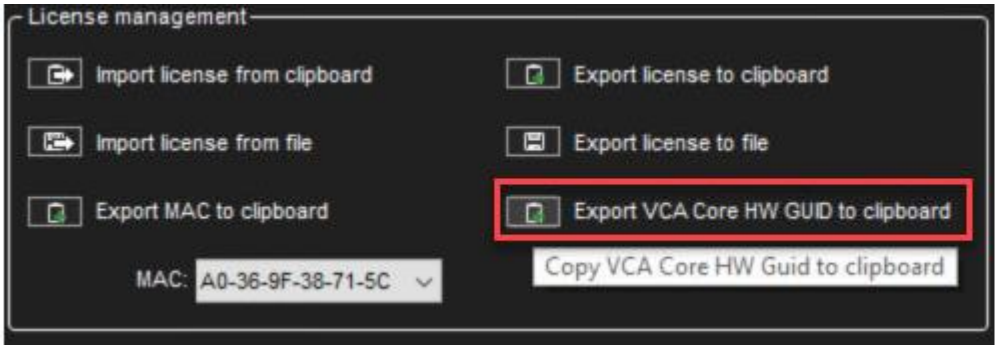
- Exporting VCA Core HW GUID file
- Activating VCA license
- Setting up the motion detection method
- Activating VCA Core for the cameras

### 6.3.1 Exporting VCA Core HW GUID file

1. Go to the **System** tab
2. Open **Licenses**
3. Double-click the license

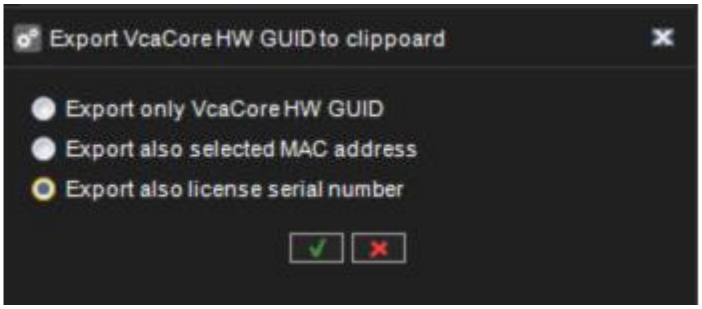


4. Select Export VCA Core HW GUID to clipboard


License management
Import license from clipboard
Export license to clipboard
Import license from file
Export license to file
Export MAC to clipboard
Export VCA Core HW GUID to clipboard
MAC: A0-36-9F-38-71-5C
Copy VCA Core HW Guid to clipboard

5. Select **Export also license serial number**

6. Click Ok


Export VcaCore HW GUID to clippoard
○ Export only VcaCore HW GUID
○ Export also selected MAC address
● Export also license serial number
✓
✗

7. Paste clipboard data to the text document.

8. Send it to Mirasys to receive the VCA license.

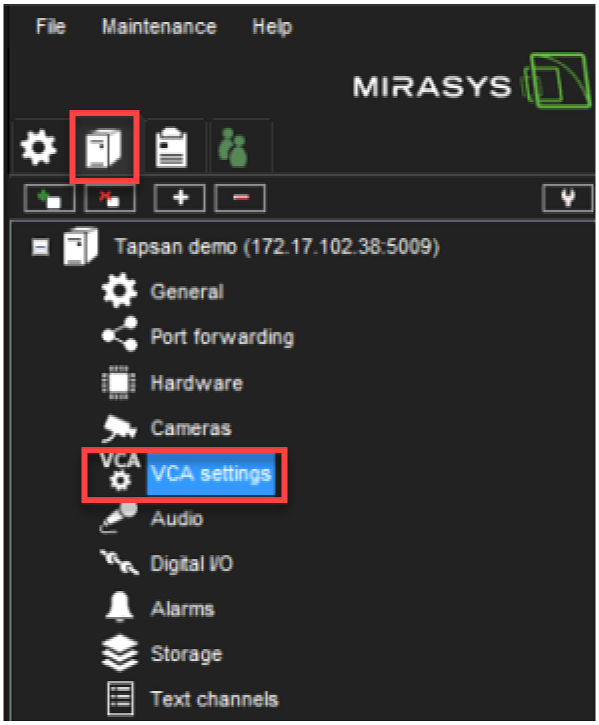
VcaCore Hw Guid: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

License serial number: YL9QMELM9QK5

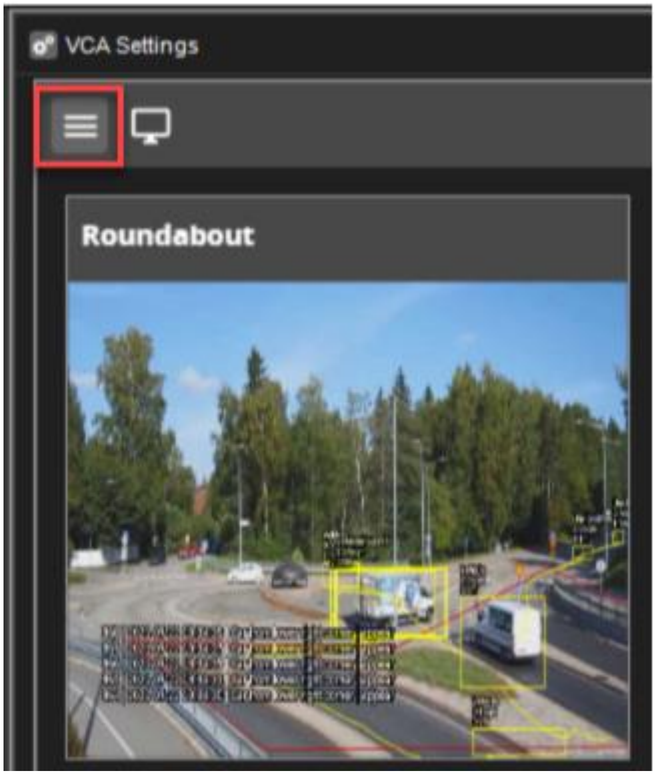
### 6.3.2 Activating VCA licenses

1. Go to the **VMS Servers** tab

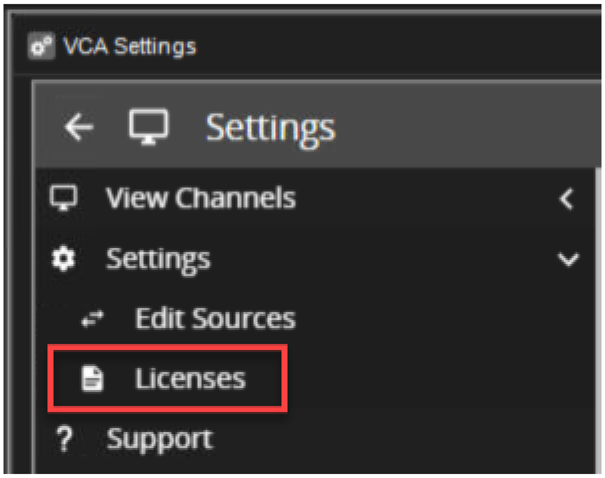
2. Open **VCA settings**



3. Click the **Settings** icon from the upper left corner

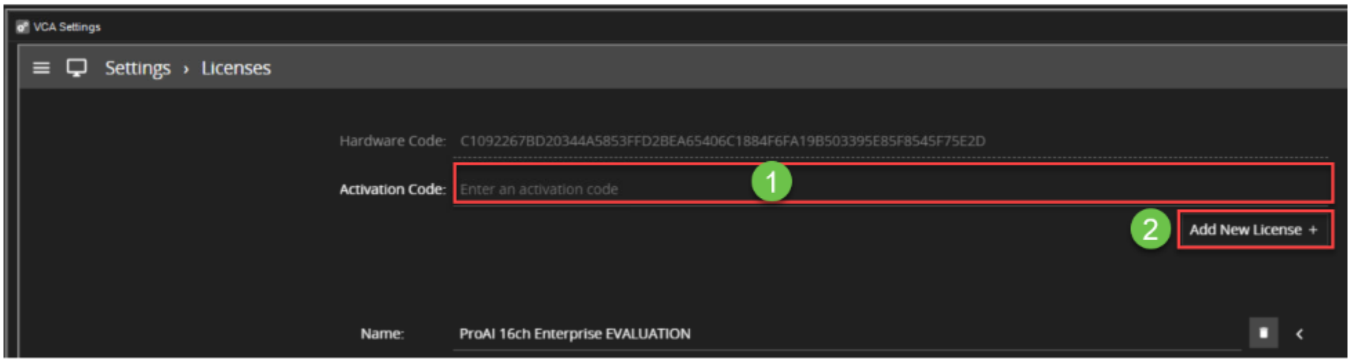


4. Select **Licenses**



5. Paste the license, which you have received from Mirasys to the **Activation Code field**

6. Click **Add New License**


VCA Settings

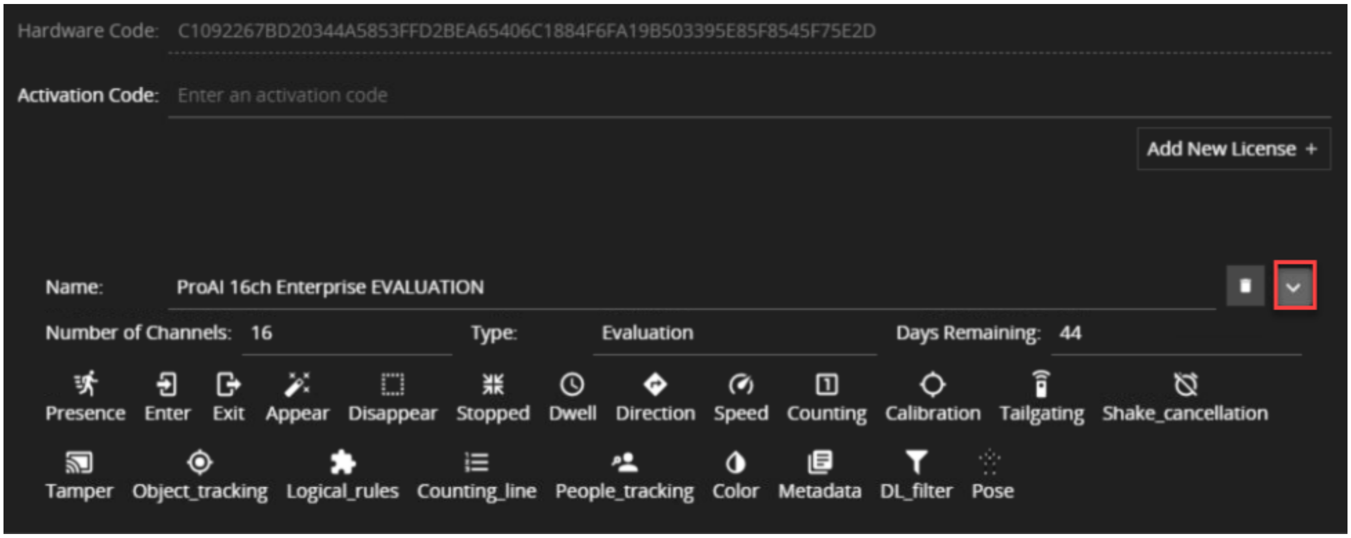
Settings › Licenses

Hardware Code: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

7. When the license is added, you will see overall information about the license and supported features


Hardware Code: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

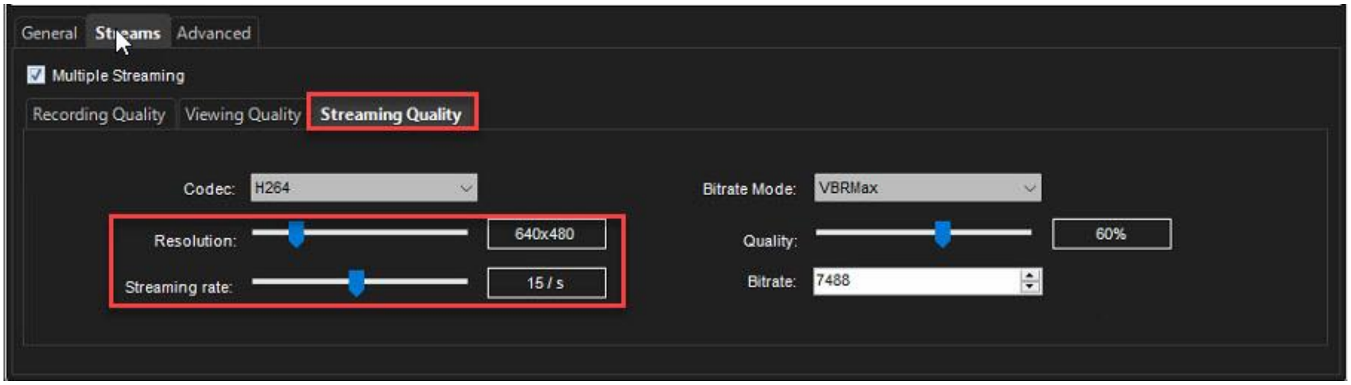
Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

Number of Channels: 16 Type: Evaluation Days Remaining: 44

### 6.3.3 Setting up the resolution and record rate

1. Enable **Multiple Streaming** to the cameras which are used for the Mirasys VCA
2. Set **Streaming Quality** resolution to **640x480**
3. Set **Streaming Quality** Streaming rate to **15/s**



☑ Multiple Streaming

Codec: H264

Resolution: 640x480

Streaming rate: 15 / s

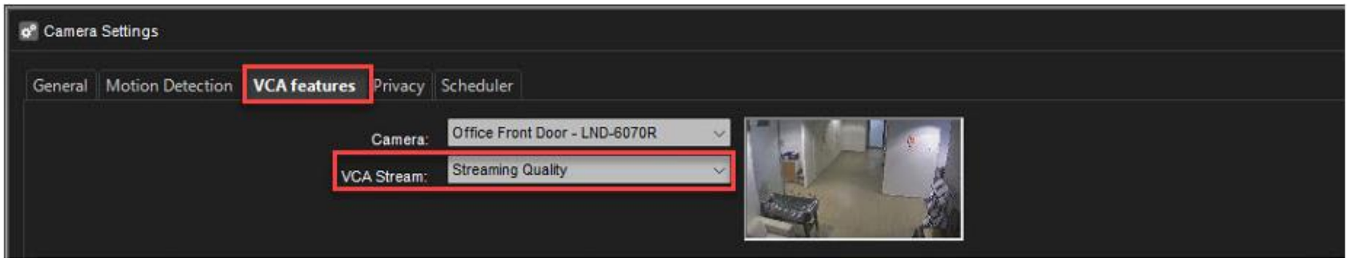
Bitrate Mode: VBRMax

Quality: 60%

Bitrate: 7488

### 6.3.4 Selecting VCA Stream

1. Open **VCA Features** tab
2. Select camera from the list
3. Open **VCA Stream** dropdown list and select **Streaming Quality**

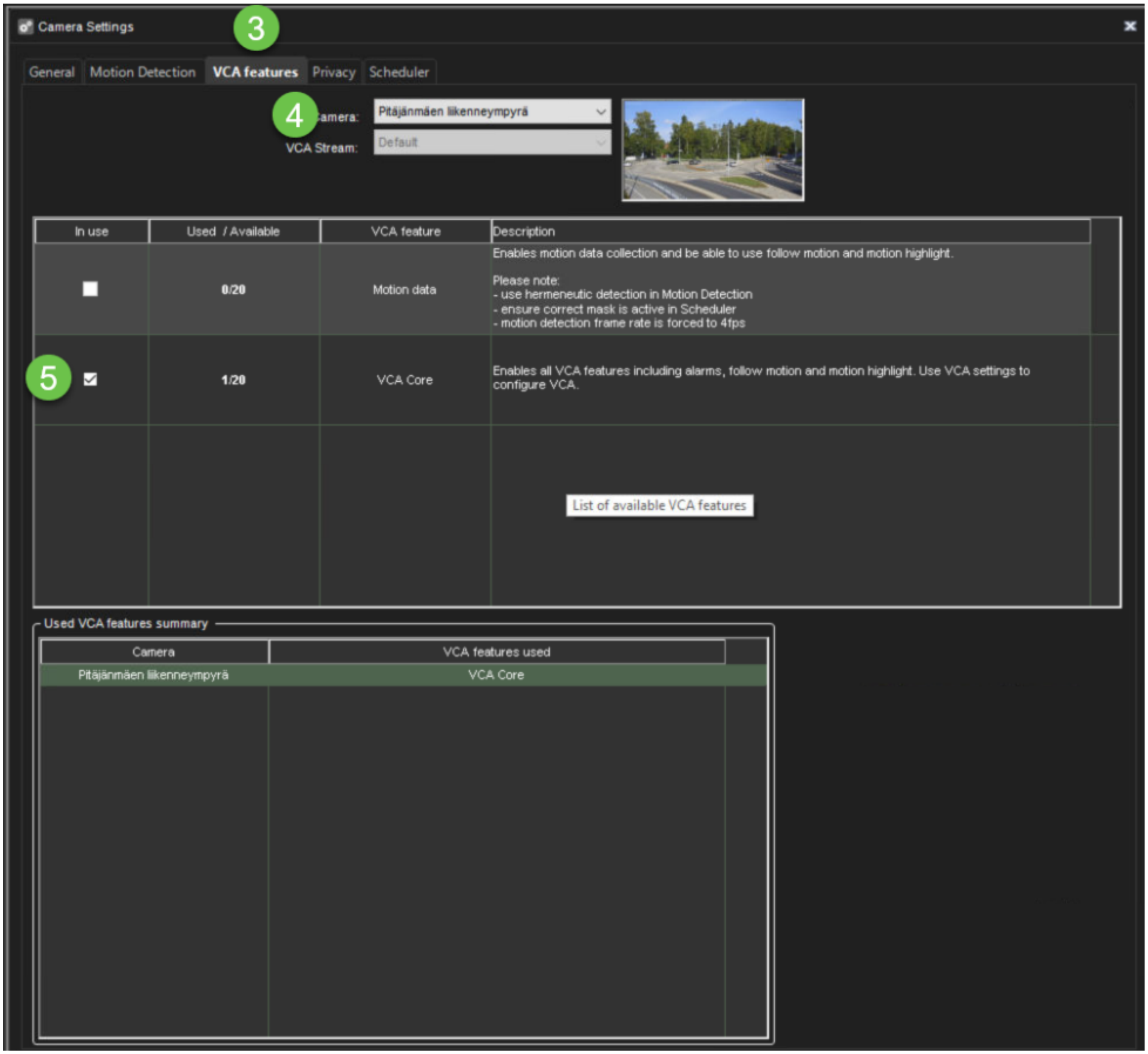


Camera: Office Front Door - LND-6070R

VCA Stream: Streaming Quality

### 6.3.5 Activating VCA Core for the cameras

1. Go to the **VMS servers tab**
2. Open **Cameras**
3. Select **VCA features**
4. Select camera
5. Enable **In use**
6. Click **OK**



Camera: Pitäjänmäen liikenneympyrä

VCA Stream: Default

| In use | Used / Available | VCA feature | Description |
|---|---|---|---|
| □ | 0/20 | Motion data | Enables motion data collection and be able to use follow motion and motion highlight.<br>Please note:<br>- use hermeneutic detection in Motion Detection<br>- ensure correct mask is active in Scheduler<br>- motion detection frame rate is forced to 4fps |
| ☑ | 1/20 | VCA Core | Enables all VCA features including alarms, follow motion and motion highlight. Use VCA settings to configure VCA. |

List of available VCA features

Used VCA features summary

| Camera | VCA features used |
|---|---|
| Pitäjänmäen liikenneympyrä | VCA Core |

## 6.4 Supported operating systems

Mirasys use third party solution for VCA which is integrated to Mirasys VMS.

Offically supported operating system are

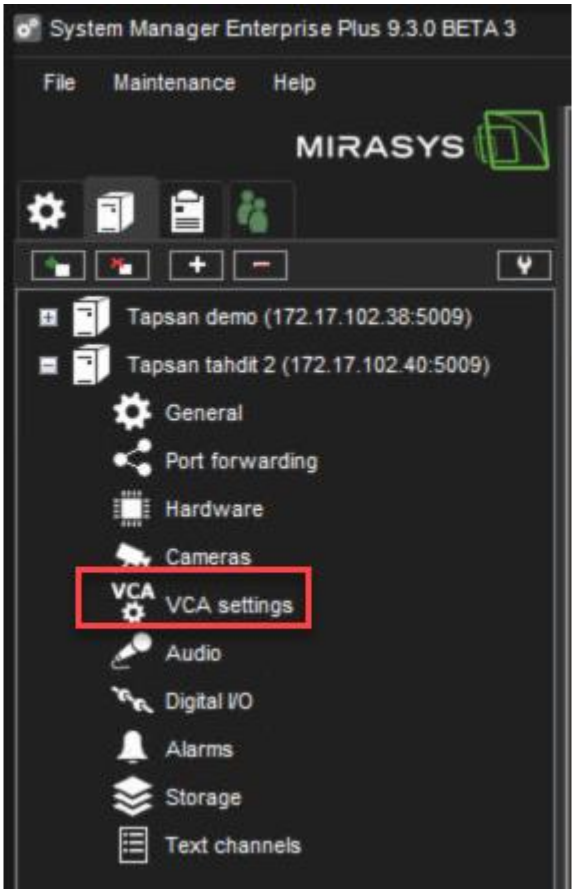
- Windows 10
- Ubuntu 18.04

With these operating system you get best performance out.

You can still use any other newer operating system or server operating system for Mirasys VMS server.

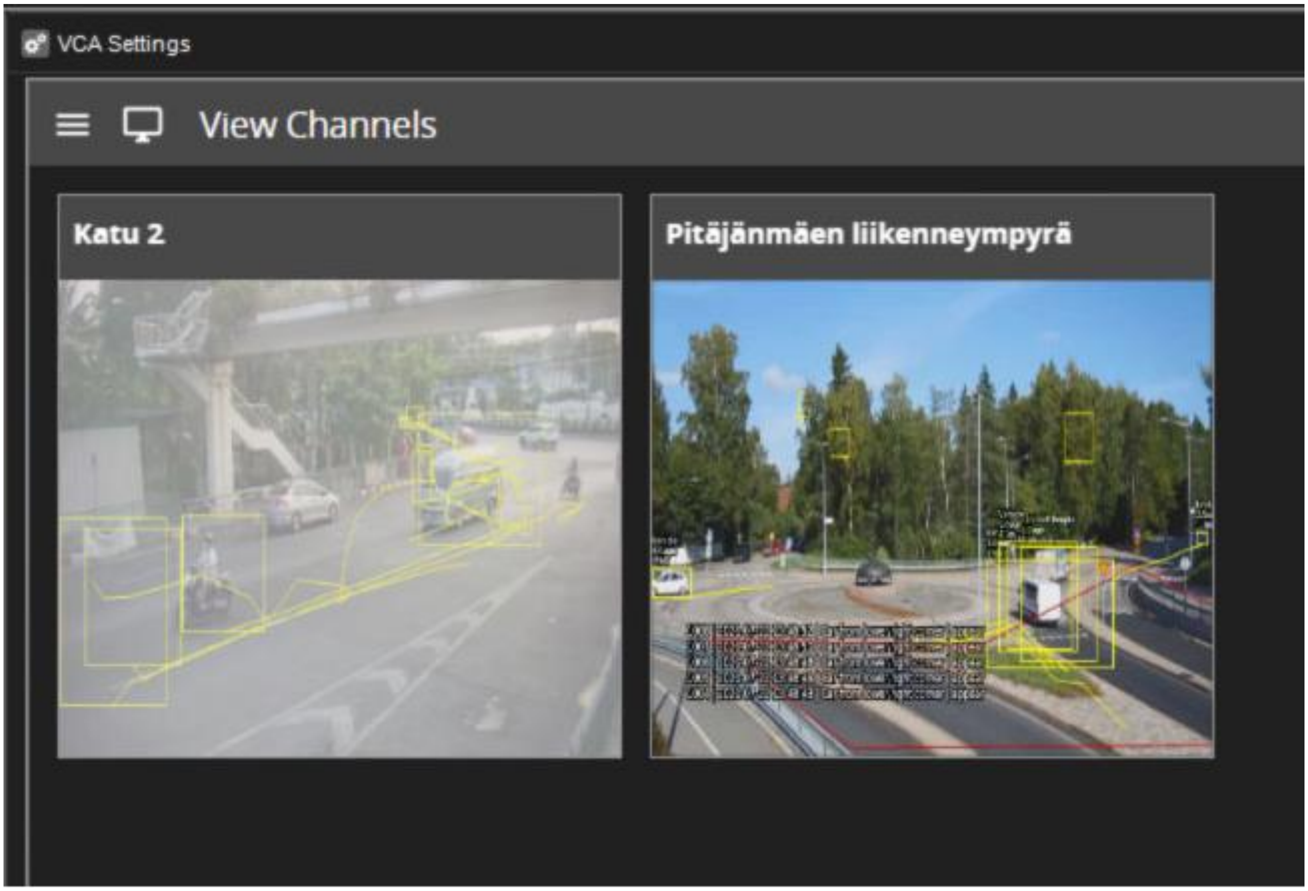
## 6.5 VCA Settings on System Manager

1. Go to the **VMS Server** tab
2. Open **VCA Settings**



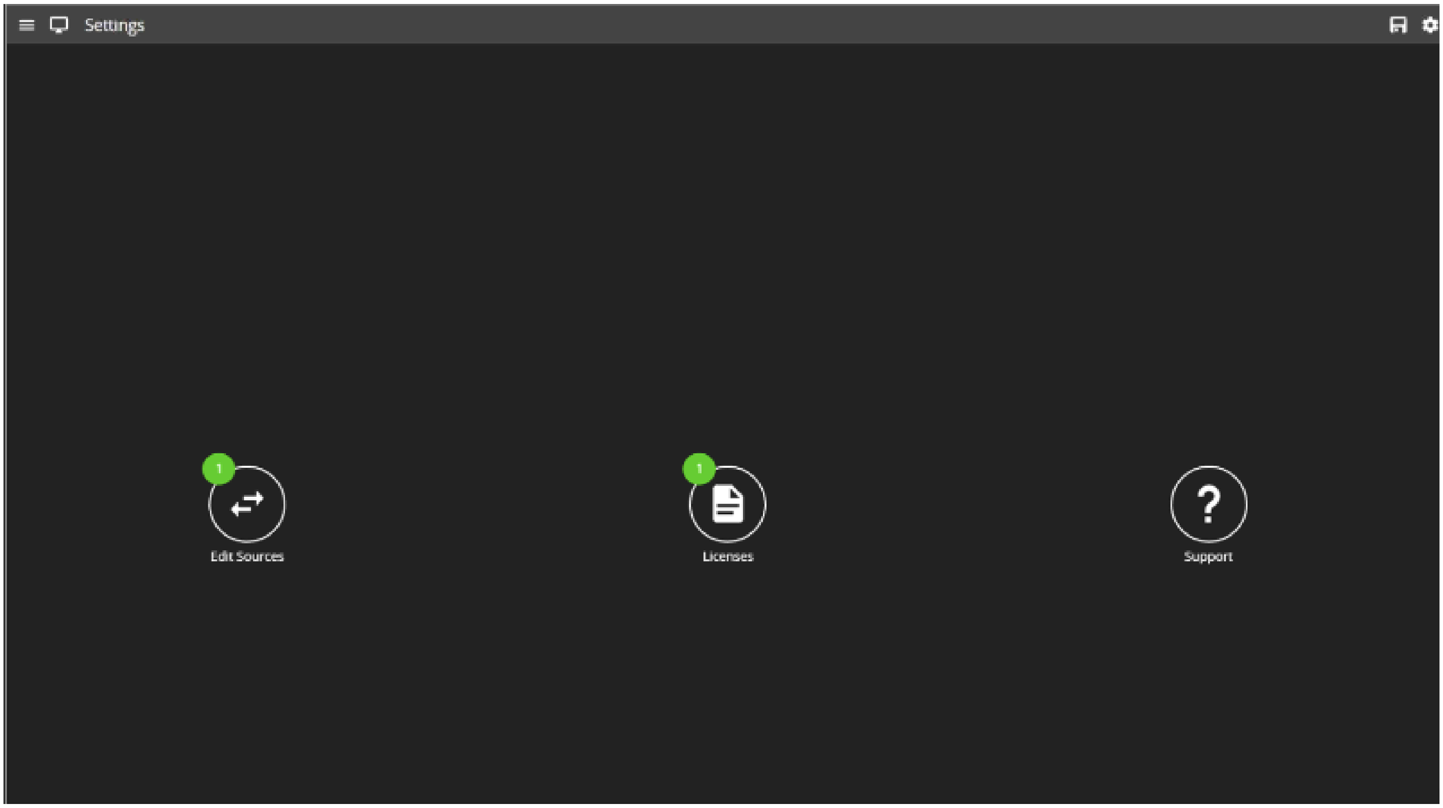
### 6.5.1 View Channels

View Channels show all cameras, which **VCA Core** is enabled.



### 6.5.2 Settings

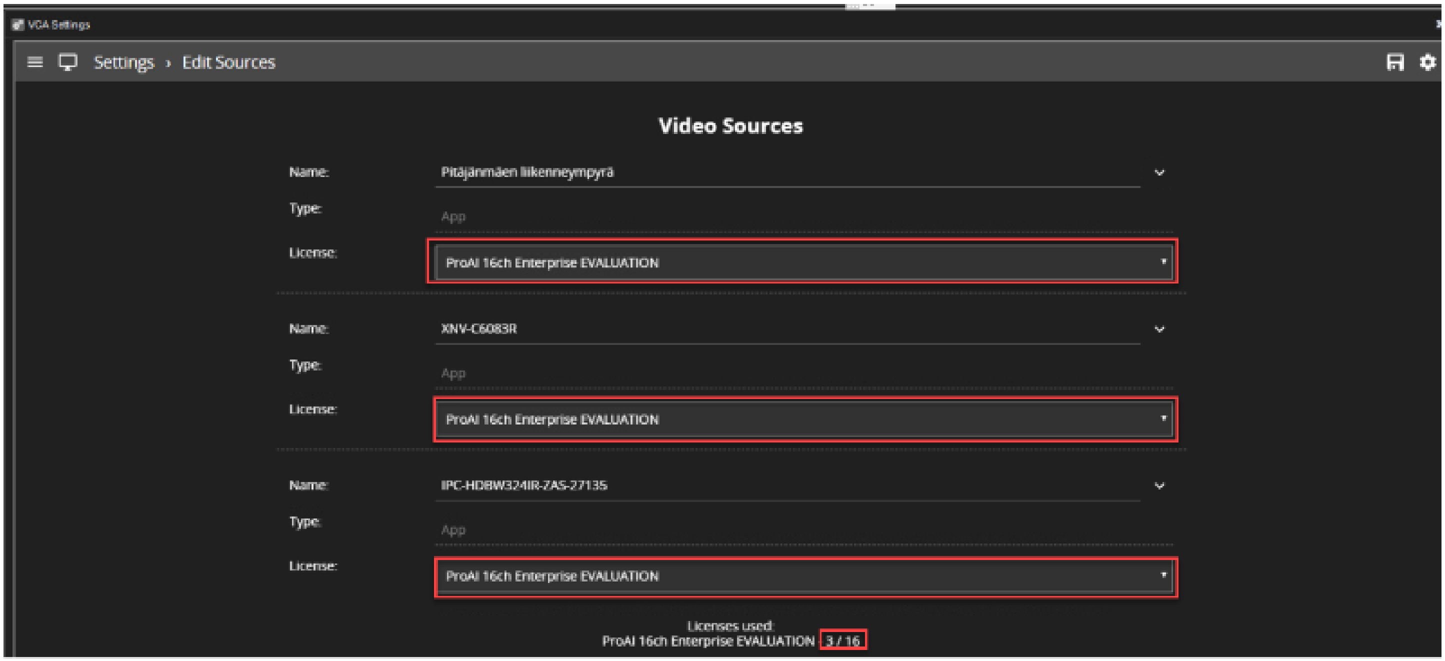
- Edit Sources
- Licenses
- Support



### 6.5.3 Edit Sources

Edit Sources shows which cameras have been used for the VCA Core.

Users can also see the type of the VCA license and how many channels have been used for the VCA license.



Video Sources

Name: Pitäjänmäen liikenneympyrä
Type: App
License: ProAI 16ch Enterprise EVALUATION

Name: XNV-C6083R
Type: App
License: ProAI 16ch Enterprise EVALUATION

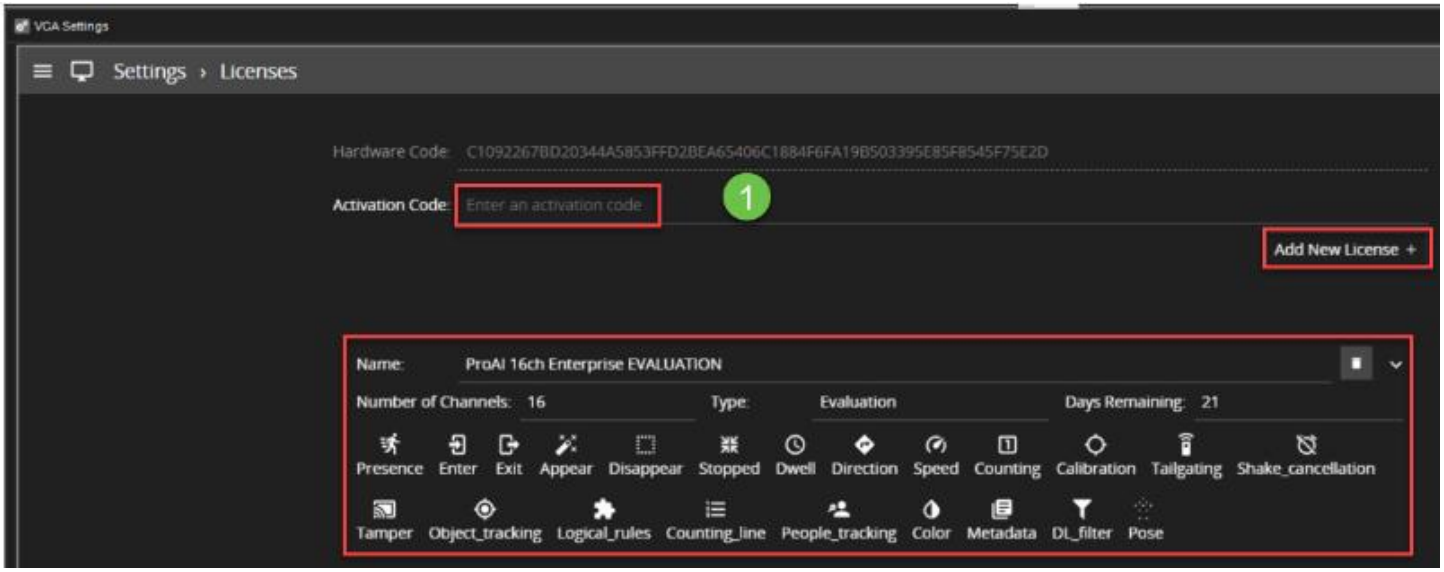
Name: IPC-HDBW3241R-ZAS-27135
Type: App
License: ProAI 16ch Enterprise EVALUATION

Licenses used:
ProAI 16ch Enterprise EVALUATION 3 / 16

### 6.5.4 Licenses

Licenses show existing license type and that features

Users can add more VCA licenses



Hardware Code: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

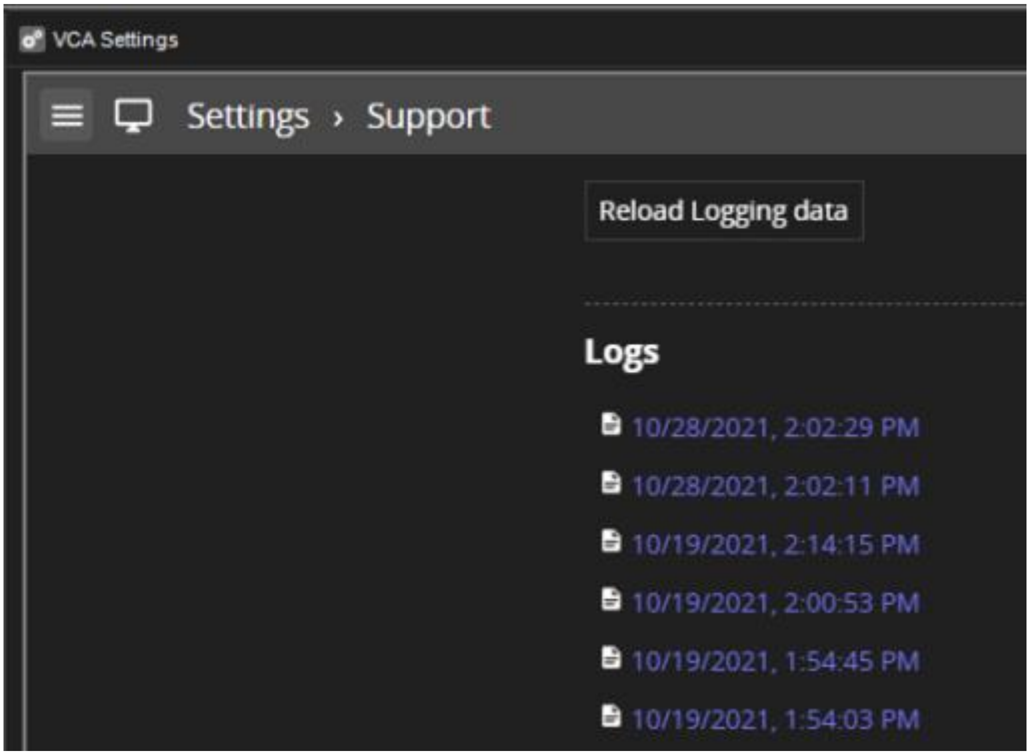
Number of Channels: 16 Type: Evaluation Days Remaining: 21

Presence Enter Exit Appear Disappear Stopped Dwell Direction Speed Counting Calibration Tailgating Shake_cancellation

Tamper Object_tracking Logical_rules Counting_line People_tracking Color Metadata DL_filter Pose

### 6.5.5 Support

Support show logs from the Mirasys VCA and users can download logs



Logs

- 10/28/2021, 2:02:29 PM
- 10/28/2021, 2:02:11 PM
- 10/19/2021, 2:14:15 PM
- 10/19/2021, 2:00:53 PM
- 10/19/2021, 1:54:45 PM
- 10/19/2021, 1:54:03 PM

## 6.6 Mirasys VCA Deep Learning

### 6.6.1 Requirements

- Nvidia GPU with CUDA cores
  - A NVIDIA GPU with CUDA Compute Capability 7.5 or higher
  - Depending on GPU CUDA cores, how many Deep Learning channels you can use on the system
- The latest NVIDIA graphics drivers (at least 460.73 or higher).
- CUDA Toolkit
- Mirasys VMS 9.4 or newer
- Deep Learning object files

### 6.6.2 Installation

1. Install latest Nvidia drivers to the system
2. Download **Mirasys VCA Deep Learning** package from Mirasys Extranet
3. Extract the package
4. Browse to folder **CUDA Toolkit**
5. Install **CUDA Toolkit** with all features
    a. You can find detailed installation guide here.
    b. Some features are not installed because Microsoft Visual Studio is not needed to install but the toolkit is providing example files
    c. If you have installed already Mirasys VMS, before copying files VMS services need to stop
6. Stop services: **WDServer**, **DVRServer** and **SMServer**
    a. This is not needed to do if you are using V9.6 or newer
7. Copy the content of the **VCA Deep Learning file**s folder to **C:\Program Files\DVMS\DVR\vca\bin** location
    a. This is not needed to do if you are using V9.6 or newer

This path is the default installation location of Mirasys VMS
If you have installed Mirasys VMS to another location, copy files there

1. Start **WDServer**, **DVRServer** and **SMServer** services
    a. This is not needed to do if you are using V9.6 or newer

Now you have installed and are ready to go with Deep Learning tracking.

Licensing is done via local VCA Deep Learning licensing or using License Server (Virtual Environment or if you want to handle licenses in one place).

Some cases detection may not work correctly. Please try to increase image quality or move/zoom camera image to closer wanted detection area.
Models are trained using clear images and some cases when using black/white image or thermal camera image this may cause that detection is working correctly. For this you can try use Deep Learning Filter with Object Tracker.

## 6.7 Mirasys VCA License Server

This license server allows the use of VCA in virtual machine/s or if you want to handle licensing in one place for all servers. For this, you need to install Mirasys VCA License Server to physical hardware and license it.
This server can then share licenses to virtual machine/s.

This feature is supported by 9.4 forward.

Do not install any other services to license server. This can cause issues on licensing side.

You don't need install VCA License Server if you are using example Master as License Server and this is physical server. On this case VMS include VCA and you can install licenses via System Manager - VCA Settings. Then connect each servers to this Master IP-address.

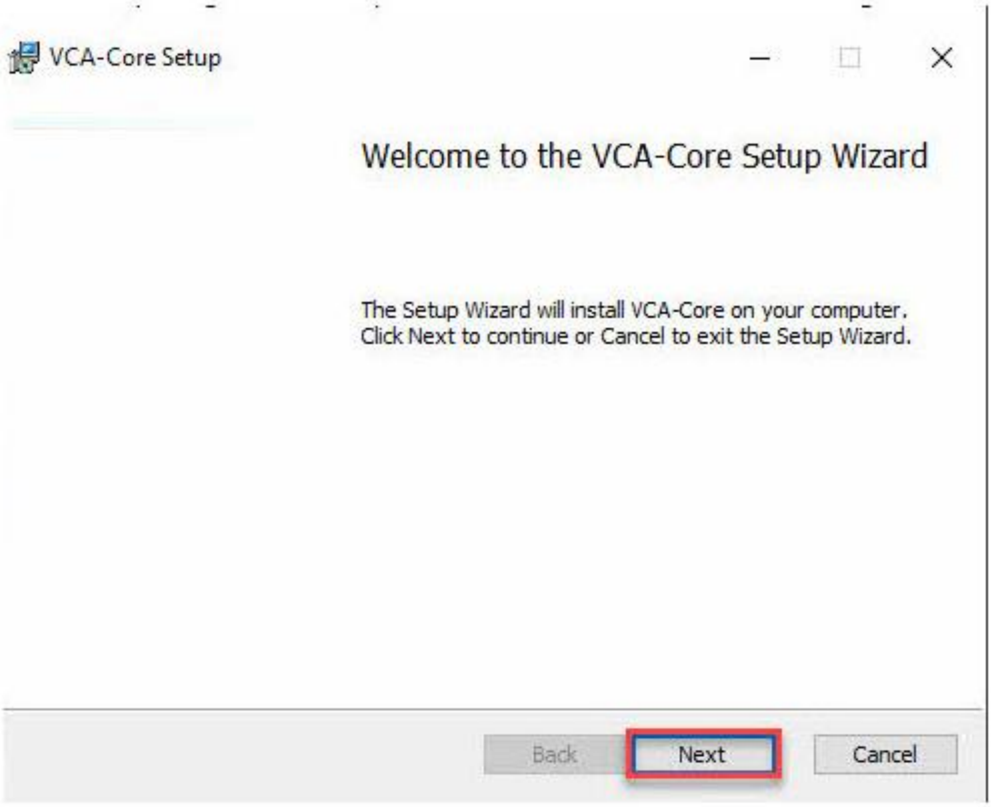
### 6.7.1 Port

- 8080, TCP for VCA License Server Management
- 15769, TCP for VCA License Port

### 6.7.2 Installation

1. Download the latest Mirasys VCA License Server package from Mirasys Extranet.
2. Extract ZIP-package on the wanted place and start installation double-clicking installation file

3. Click **Next** to proceed

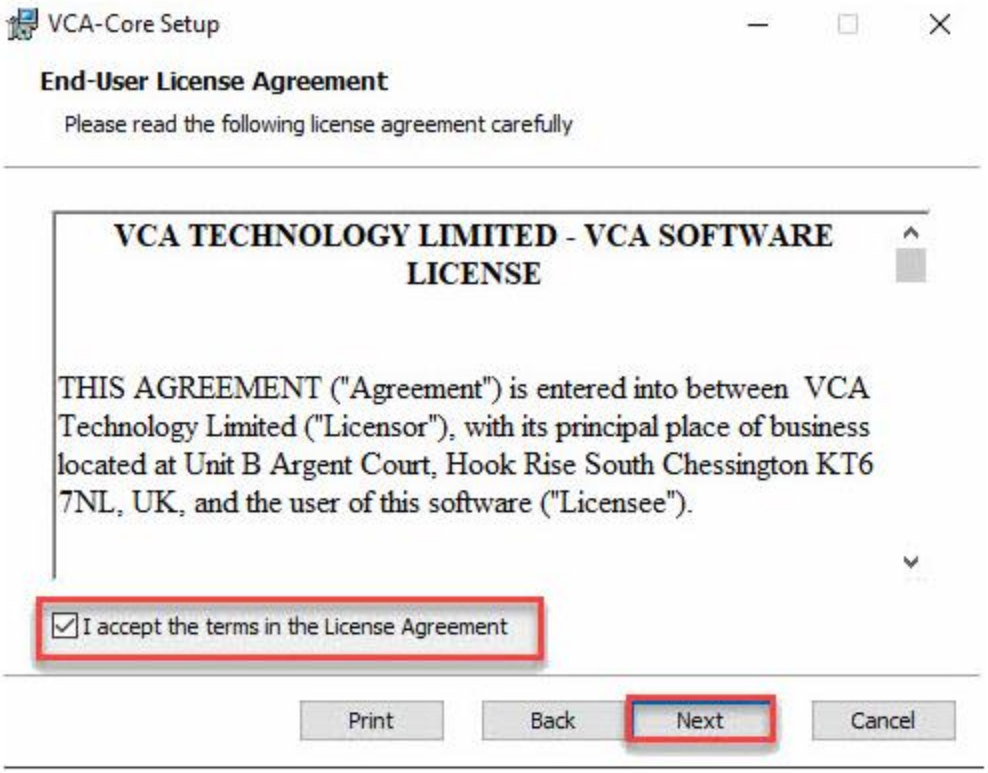


VCA-Core Setup

Welcome to the VCA-Core Setup Wizard

The Setup Wizard will install VCA-Core on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.

4. Accept **End-User License Agreemen**t and click **Next**


VCA-Core Setup
End-User License Agreement
Please read the following license agreement carefully

VCA TECHNOLOGY LIMITED - VCA SOFTWARE LICENSE

THIS AGREEMENT ("Agreement") is entered into between VCA Technology Limited ("Licensor"), with its principal place of business located at Unit B Argent Court, Hook Rise South Chessington KT6 7NL, UK, and the user of this software ("Licensee").

☑ I accept the terms in the License Agreement

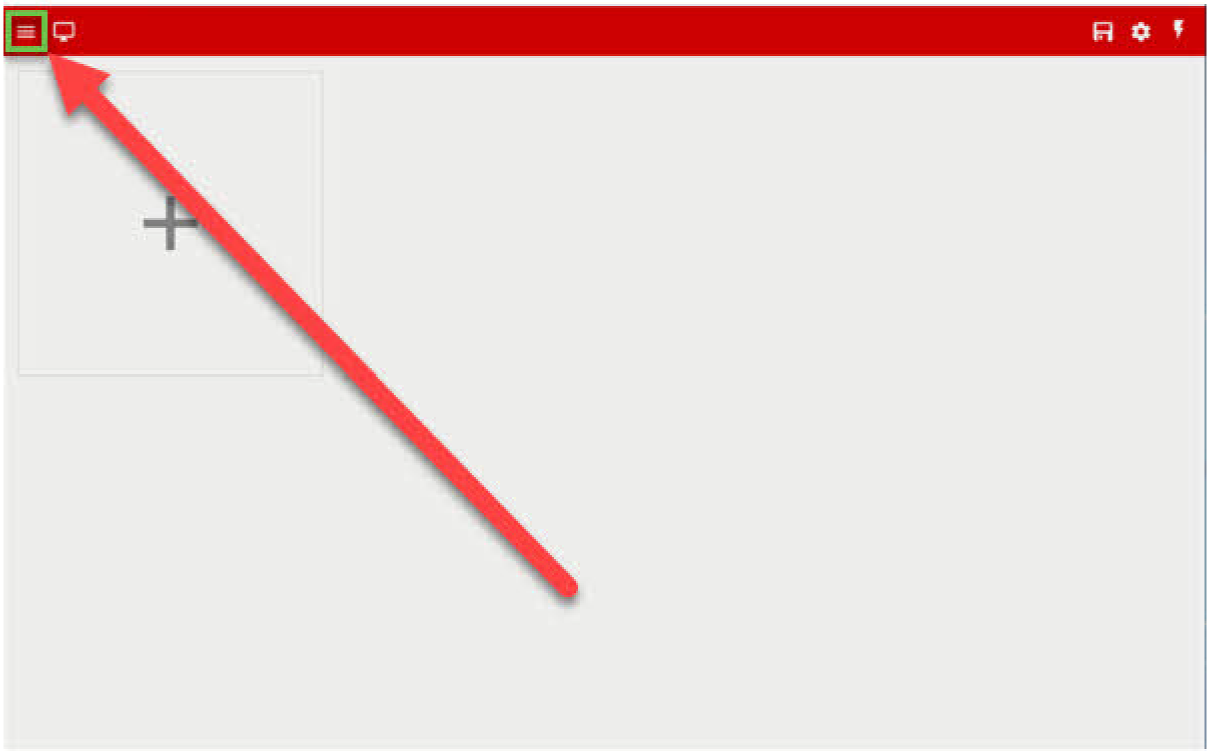
5. Follow instructions until the installation is finalized

### 6.7.3 Usage and licensing

To log in to Mirasys VCA License Server, you need to use the browser and go to the site http://localhost:8080/.

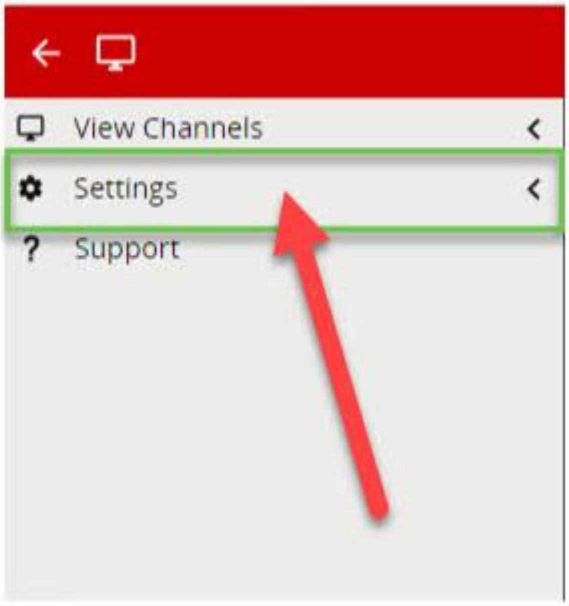
The default username is **admin,** and the default password is **admin**

### 6.7.3.1 On the main page, you can access settings via the burger menu.

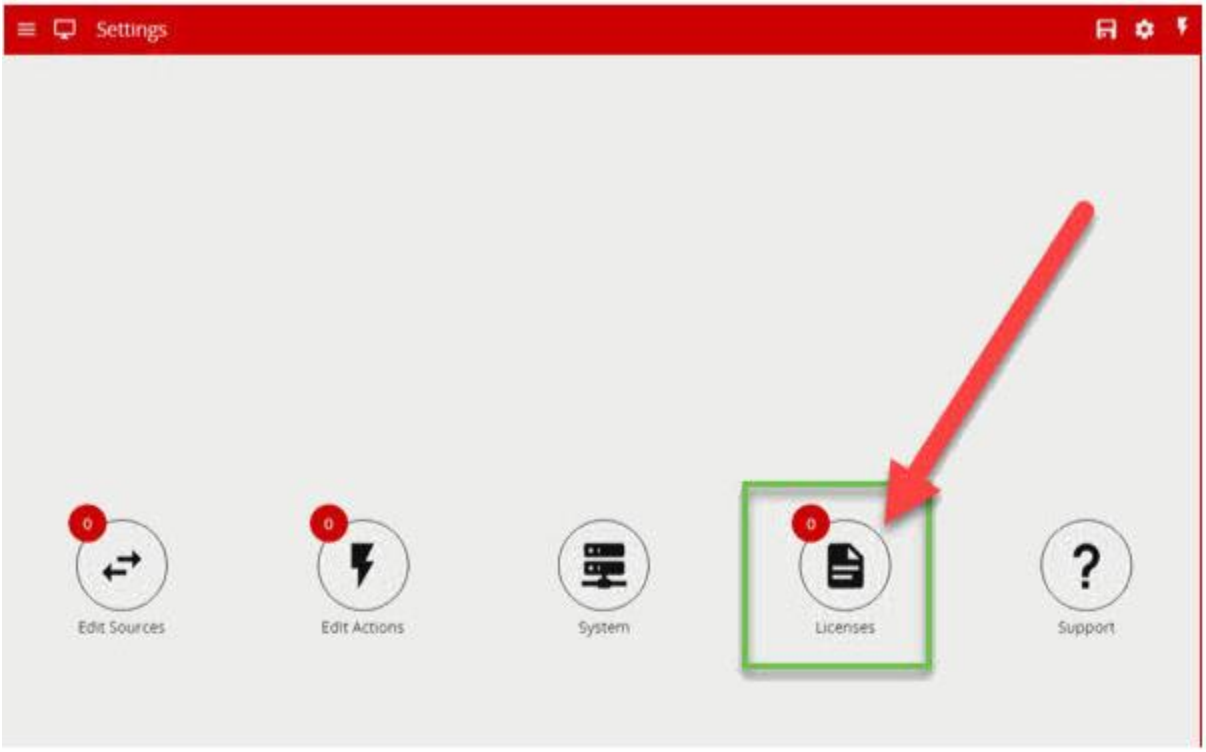


### 6.7.3.2 Adding the license

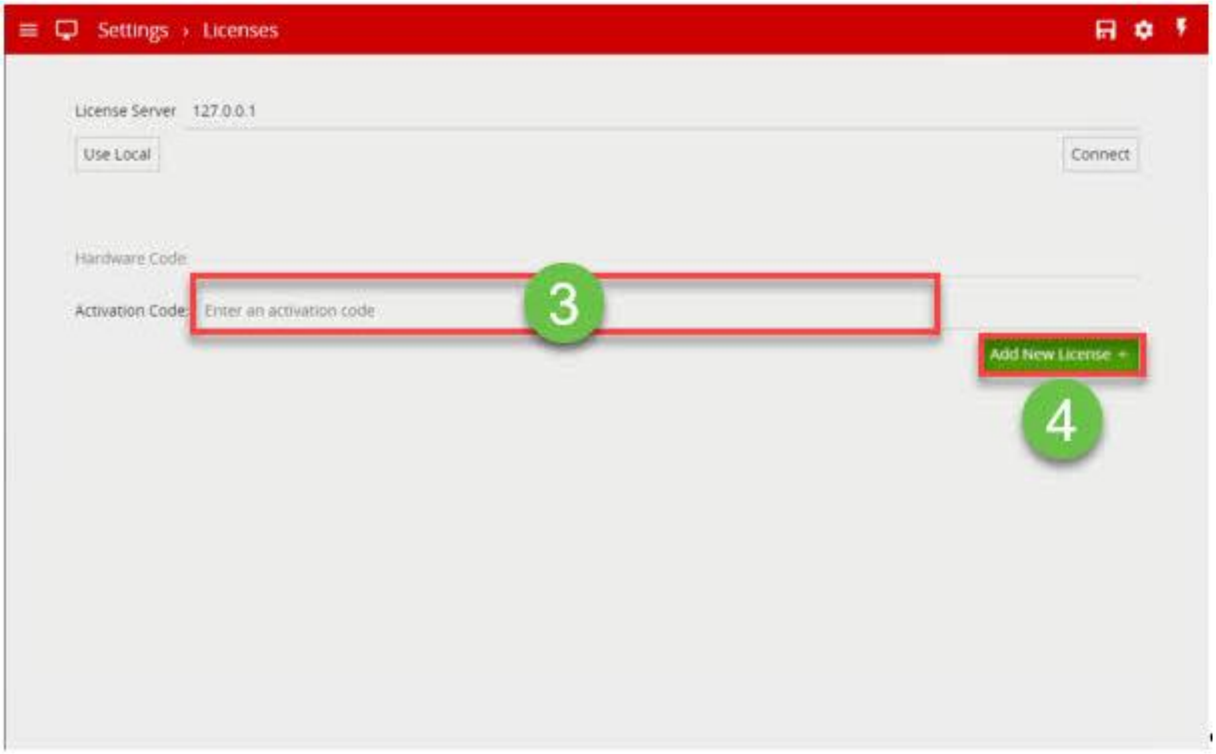
1. Open **Settings**



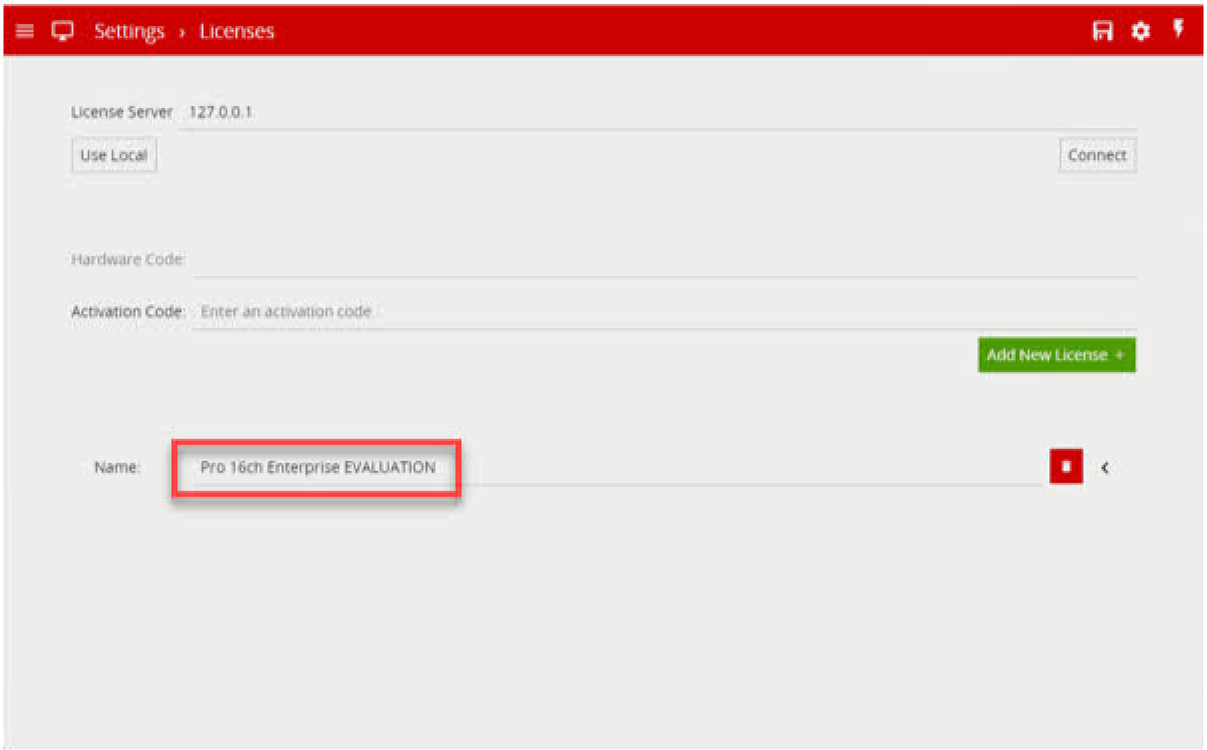
2. Open **Licenses**



3. Copy Hardware Code and send it to Mirasys to receive license details.
4. When you have received the activation code from Mirasys, paste the code to the **Activation Code** field and click **Add the new license**



5. When you have added wanted licenses or licenses to the system, you can proceed on Mirasys VMS side.



License Server 127.0.0.1

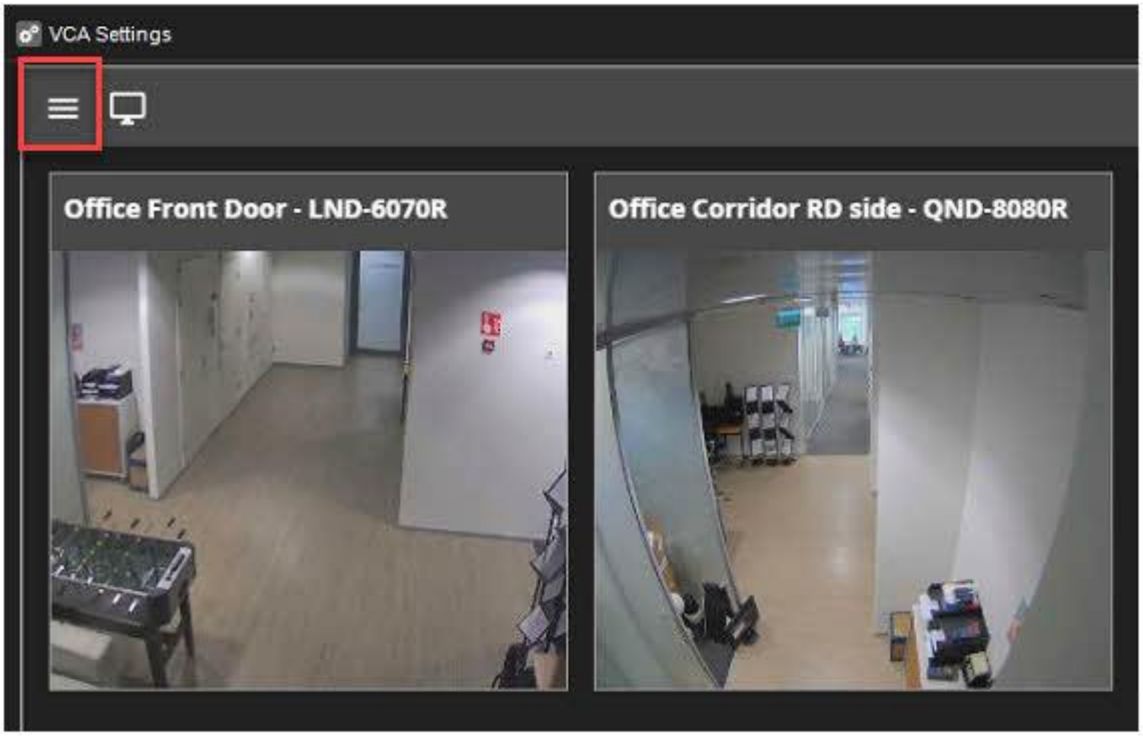
Hardware Code:

Activation Code: Enter an activation code

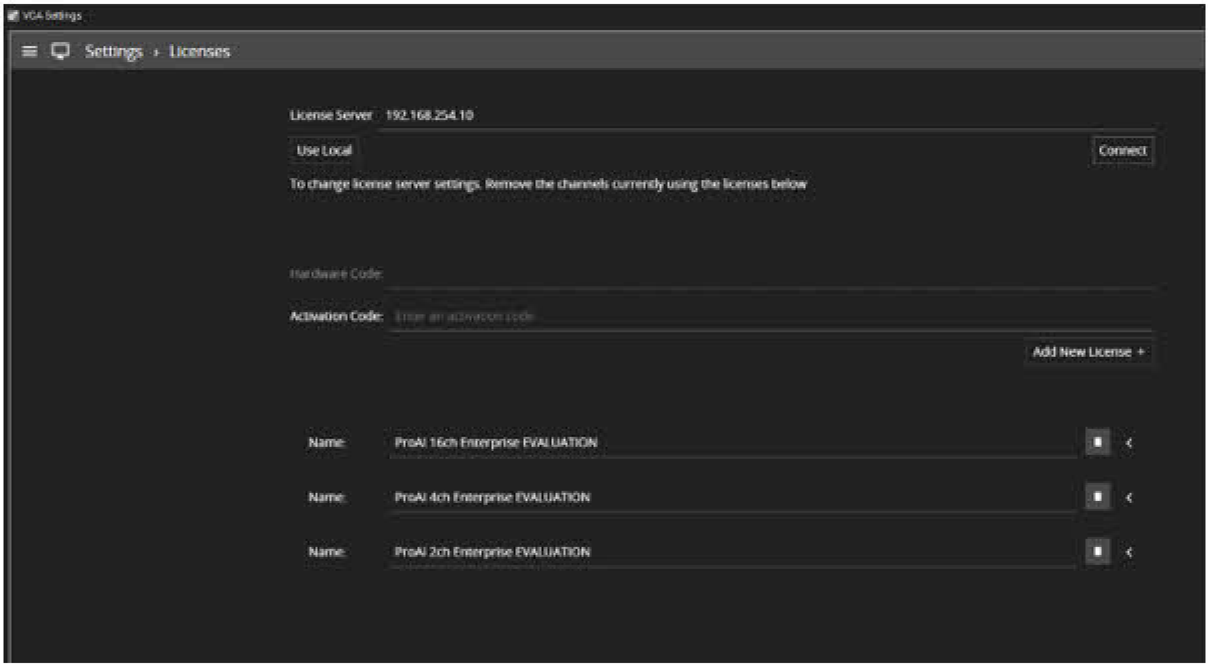
Name: Pro 16ch Enterprise EVALUATION

### 6.7.4 Mirasys VMS Configuration

1. Open System Manager, go to the server section and select VCA Settings.
2. This opens a new window to find a similar burger menu like earlier.


VCA Settings
Office Front Door - LND-6070R
Office Corridor RD side - QND-8080R

3. Under this menu, you can find Settings to tell license server address DNS/IP-address.



License Server 192.168.254.10

To change license server settings. Remove the channels currently using the licenses below

Hardware Code:

Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

Name: ProAI 4ch Enterprise EVALUATION

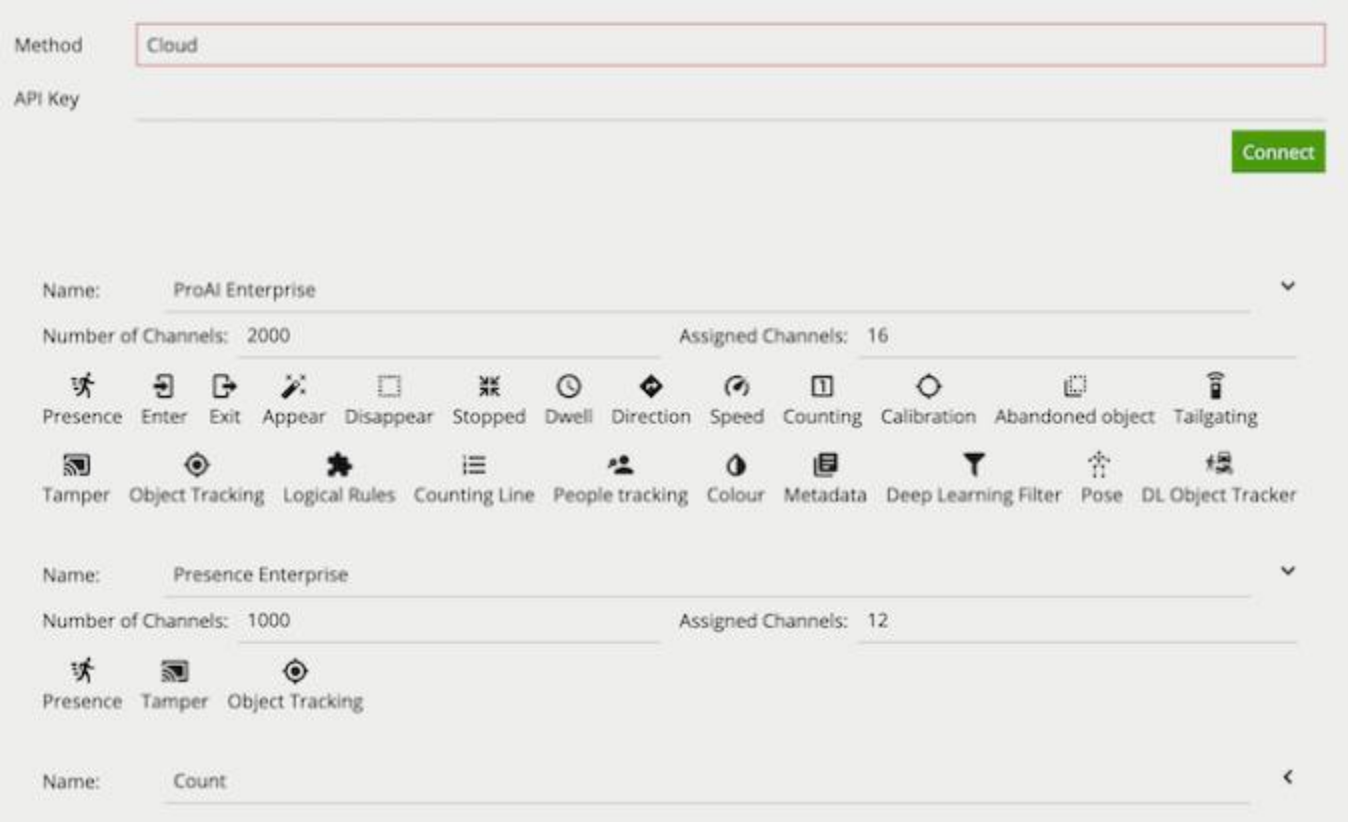
Name: ProAI 2ch Enterprise EVALUATION

4. When you have to fill License Server address, you can click Connect to this.

If the connection is made successfully, this shows Mirasys VCA License Server licenses.

After this, you can go to VCA Settings and sources to assign a wanted license to the wanted camera channel.

## 6.8 Cloud Licensing



Method: Cloud

API Key:

Name: ProAI Enterprise

Number of Channels: 2000 Assigned Channels: 16

Name: Presence Enterprise

Number of Channels: 1000 Assigned Channels: 12

Name: Count

Once a valid API Key is provided and the connection to the Cloud Licensing Server is established, the license pool associated with that API key will be shown.

When using Cloud Licensing the license pool available to VCAserver is managed using a cloud portal.

- **Method:** Switches between Cloud Licensing or a License Server.
- **API Key:** An authorization token that links to a Cloud Licensing Account.
- **Connect:** Connects this instance of VCAserver to the Cloud Licensing Servers.
- The list of installed licenses and their features are displayed underneath.

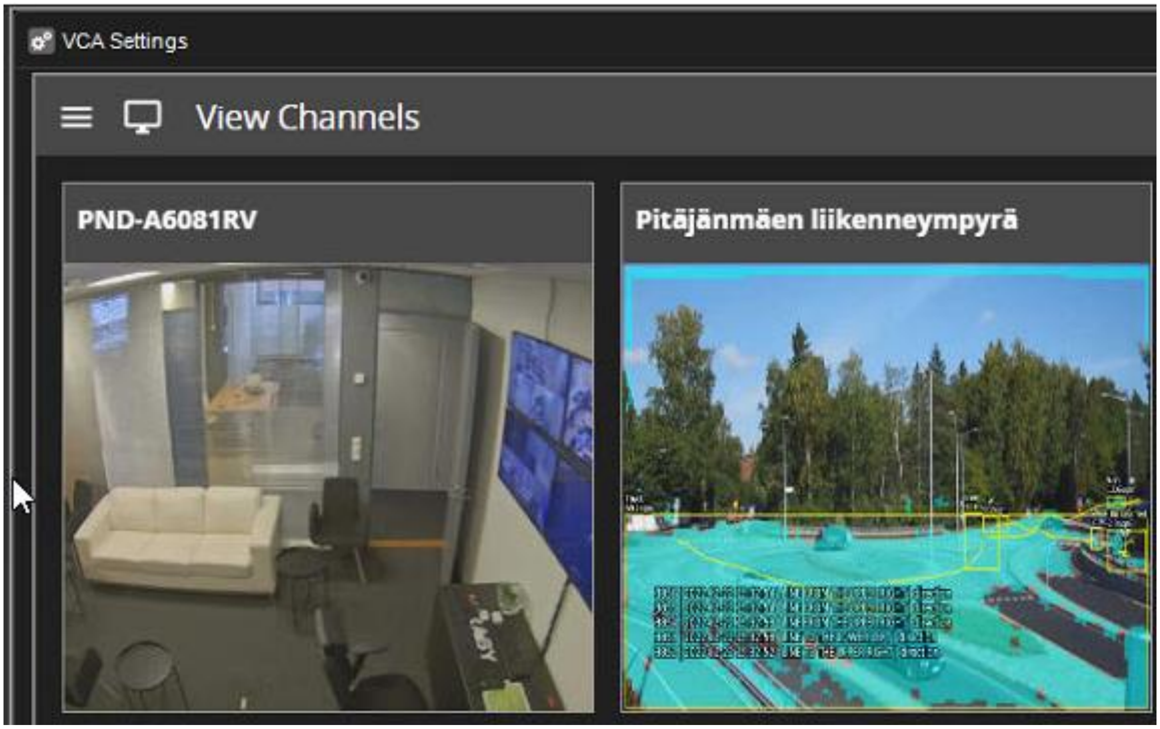
On new installations, before a user is able to add sources, the Cloud Licensing account will need a license added to the license pool.

## 6.9 VCA Channel Settings

VCA Channel Settings contains all configuration related to video channels, where VCACore is enabled.

- Tracking
- Zones
- Calibration
- Classification
- Burnt-in Annotation
- Rules
- Video Preview
- Deep-Learning

You can open under VCA Settings wanted camera by clicking it.



- VCA Tracking
- VCA Deep Learning Skeleton Tracker
- VCA Hand Object Interaction Tracker
- VCA Zones
- VCA Calibration
- VCA Classification
- VCA Burnt-in Annotation
- VCA Rules
- VCA - Deep-Learning Filter

## 6.9.1 VCA Tracking

### *6.9.1.1 Initialization*

When a tracker is selected by the user, an initialization phase will be required. This will vary based on the selected tracker.

**Object Tracker**: when selected the tracker will need to 'learn the scene' to determine background from moving foreground objects.

Whilst learning the scene the following message will be displayed in the live view, and no objects will be tracked during this time.

**DL People Tracker & DL Object Tracker**: when first selected, the DL engine will run a model generation process. This optimizes the DL models to run on the available GPU hardware.

Irrespective of which tracker is selected, the DL People tracker model, DL Object Tracker model and the DL Filter model will all be optimized in one go.

This process can take up to 10 minutes per model and may increase with different GPU configurations. Once complete the optimized models are stored in the configuration folder.

The process will not need to be run again unless the GPU hardware is changed. Whilst optimization is performed a message will be displayed in the live view, and no objects will be tracked during this time.

Please note: The DL Filter requires the same initialization process but does not display a message.

Once initialized, VCAserver will begin analyzing the video stream with the selected tracker. Settings specific to that tracker will also be displayed below the tracker engine selection option.

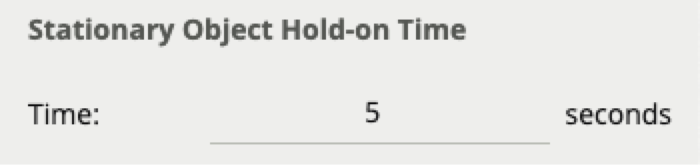
Regardless of the tracker selected, any tracked object can be passed through the available rules. However, in some cases, certain rules or algorithms will only be available with a specific tracker. For example, Deep Learning Filter and the abandoned and removed object rules are only available with the Object Tracker.

#### *6.9.1.2 Object Tracker*

The Object Tracker is a motion based detection engine. Based on changes detected in the image, the algorithm separates the image into foreground and background, tracking any foreground object that is moving above a set threshold. The Object Tracker has the following settings:

##### 6.9.1.2.1 Stationary Object Hold-on Time

The Stationary Object Hold-on Time defines the amount of time an object will be tracked by the engine once it becomes stationary.
Since objects which become stationary must be "merged" into the scene after some finite time, the tracking engine will forget about objects that have become stationary after the Stationary Object Hold-on Time.



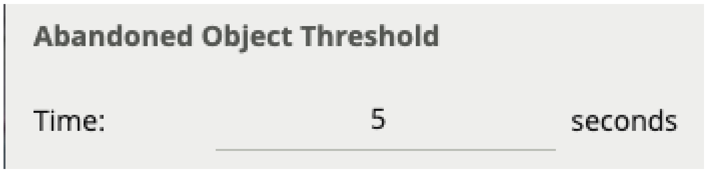
Stationary Object Hold-on Time

Time: 5 seconds

The default setting is 60 seconds.

##### 6.9.1.2.2 Abandoned / Removed Object Threshold

This threshold amount of time an object must be classed as abandoned or removed before an Abandoned / Removed rule will trigger.



Abandoned Object Threshold

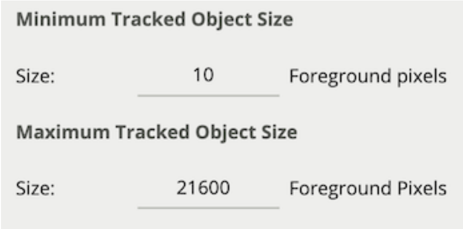
Time: 5 seconds

The default setting is 5 seconds.

### 6.9.1.2.3 Minimum Tracked Object Size

The **Minimum Tracked Object Size** defines the size of the smallest object that will be considered for tracking.

For most applications, the default setting of 10 is recommended. In some situations, where extra sensitivity is required, the value can be manually specified.
While lower values allow the engine to track smaller objects, it may increase the susceptibility to false detections.



Minimum Tracked Object Size

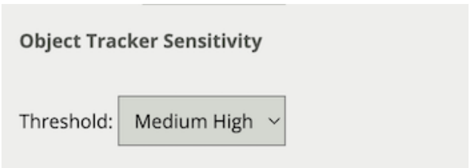
Size: 10 Foreground pixels

Maximum Tracked Object Size

Size: 21600 Foreground Pixels

### 6.9.1.2.4 Object Tracker Sensitivity

The **Object Tracker Sensitivity** value allows the object tracker to be tuned to ignore movement below a certain threshold.
Combined with the Blob Map burnt in annotation, which visualizes the area of the scene the object tracker is detecting movement, this value can be adjusted to filter out environmental noise.



Object Tracker Sensitivity

Threshold: Medium High

The default setting is 4.

### 6.9.1.2.5 Scene Change Detection (Object Tracker)

Learn more about Scene Change Detection.

### 6.9.1.2.6 Detection Point of Tracked Objects

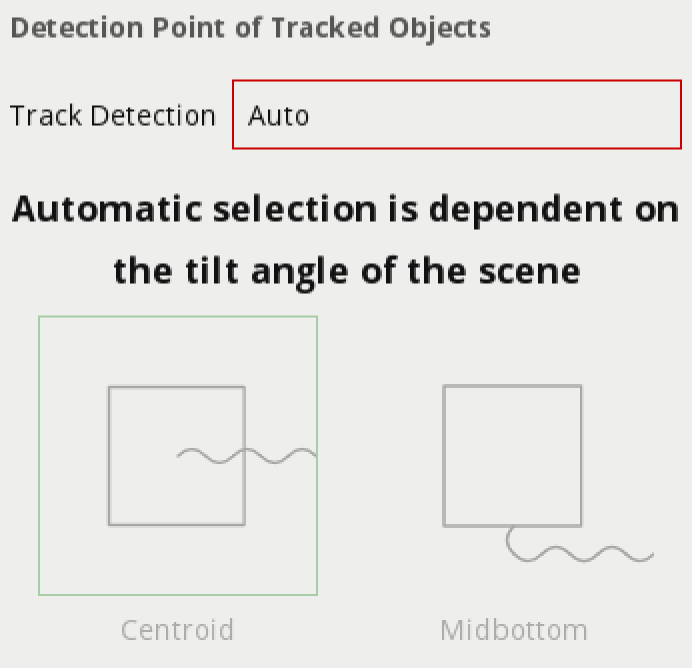
For every tracked object, a point is used to determine the object's position, and evaluate whether it intersects a zone and triggers a rule. This point is called the detection point.

There are 3 modes that define the detection point relative to the object:

#### *6.9.1.2.6.1 Automatic*

In automatic mode, the detection point is automatically set based on how the channel is configured.

It selects 'Centroid' if the camera is calibrated overhead, or 'Mid-bottom' if the camera is calibrated side-on or not calibrated.



Detection Point of Tracked Objects

Track Detection: Auto

Automatic selection is dependent on the tilt angle of the scene

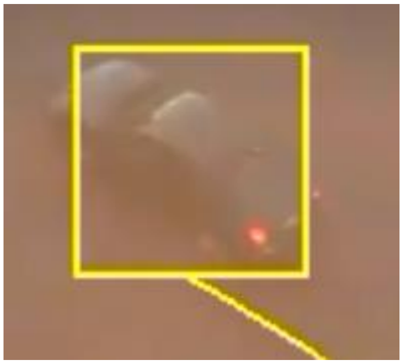
#### *6.9.1.2.6.2 Centroid*

In this mode, the detection point is forced to be the centroid of the object.

##### *6.9.1.2.6.3 Mid-bottom*

In this mode, the detection point is forced to be the middle of the bottom edge of the tracked object.

Normally this is the ground contact point of the object (where the object intersects the ground plane).



#### 6.9.1.2.7 Tamper Detection (Object Tracker)

Learn more about Tamper Detection.

#### 6.9.1.2.8 Loss Of Signal Emit Interval

See Loss Of Signal Emit Interval

### ***6.9.1.3 Deep Learning People Tracker***

The Deep Learning People tracker is designed to track people in situations where the camera field of view is relatively close.

The Deep Learning People Tracker is based on Pose Estimation technology, providing the location of a person in the field of view as well as additional key point metadata on the parts of the body. See Deep Learning Requirements for hardware requirements for this algorithm.

The Deep Learning People Tracker has the following settings:

#### 6.9.1.3.1 Tamper Detection (DLPT)

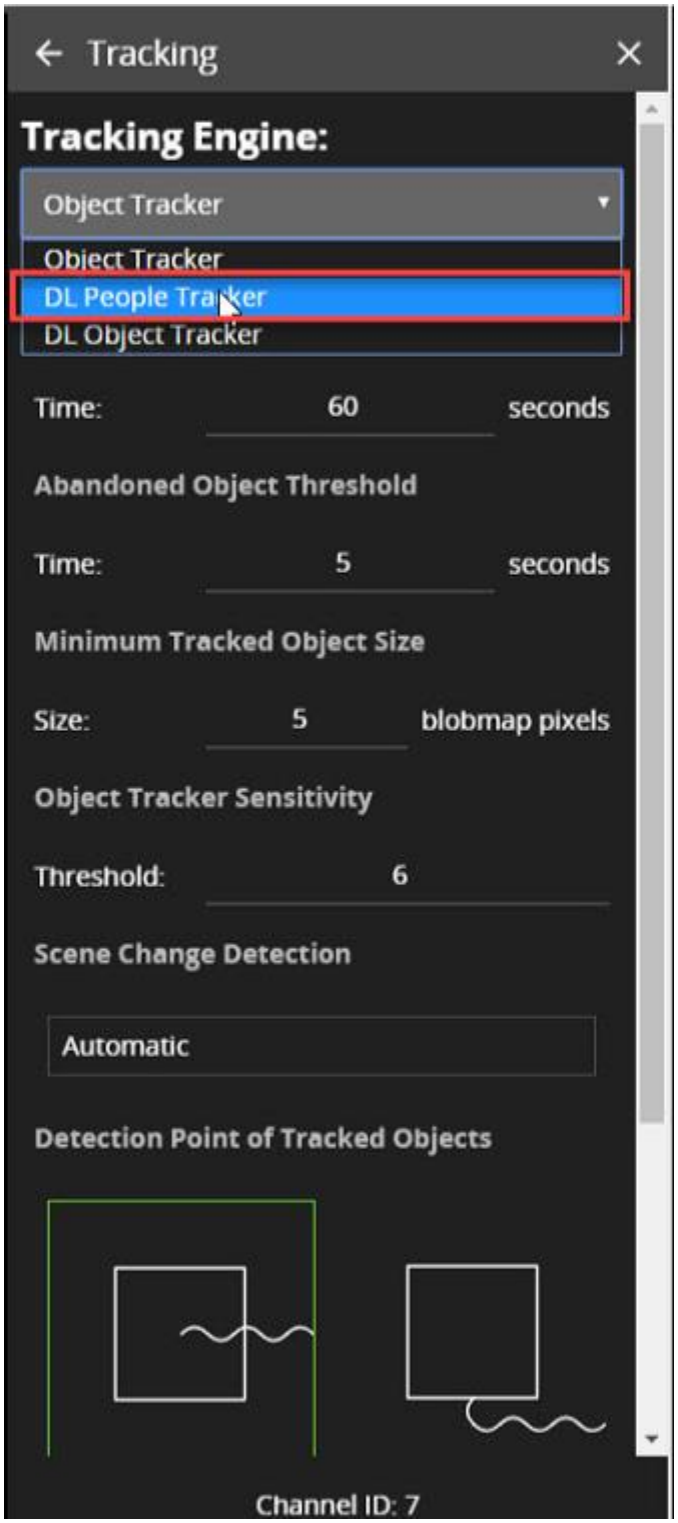
Learn more about Tamper Detection.

#### 6.9.1.3.2 Loss Of Signal Emit Interval

See Loss Of Signal Emit Interval

#### 6.9.1.3.3 Enabling DL People Tracker

1. Open **View Channels**
2. Select the camera
3. Open **Tracking**
4. Open **Tracking Engine** dropdown box and select **DL People Tracker**



Tracking Engine: Object Tracker

Time: 60 seconds

Abandoned Object Threshold

Time: 5 seconds

Minimum Tracked Object Size

Size: 5 blobmap pixels

Object Tracker Sensitivity

Threshold: 6

Scene Change Detection

Automatic

Detection Point of Tracked Objects

Channel ID: 7

#### 6.9.1.4 *Deep Learning Object Tracker*

The Deep Learning Object Tracker is designed for accurate detection and tracking of people, vehicles and key objects in challenging environments where motion based tracking methods would struggle.
The list of objects detected by the Deep Learning Object Tracker is given below:

| **Class Name** | **Description** |
|---|---|
| person | A person, or tracked object with a person present (e.g bicycle) |
| motorcycle | A motorcycle |
| bicycle | A bicycle |
| cyclist | Person riding a bicycle, can be reported as two separate objects |
| bus | A bus |
| car | A car |
| van | A van, including mini-vans and mini-buses |
| truck | A truck, including lorries and commercial work vehicles, |
| forklift | A forklift truck |
| bag | A backpack or holdall (sports bag) |
| | |

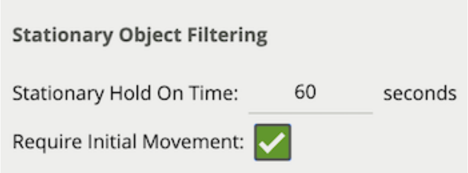
The Deep Learning Object Tracker is based on a classification and detection model, providing the location of an object in the field of view. See Deep Learning Requirements for hardware requirements for this algorithm.

The Deep Learning Object Tracker has the following settings:

##### 6.9.1.4.1 Stationary Object Filtering

See Stationary Hold On Time

In addition to the Stationary Hold On Time, an additional setting Require Initial Movement, is available which will prevent objects which have not moved from being tracked.



Stationary Object Filtering

Stationary Hold On Time: 60 seconds

Require Initial Movement: ☑

### 6.9.1.4.2 Detection Point of Tracked Objects

See Detection Point of Tracked Objects

### 6.9.1.4.3 Tamper Detection (DLOT)

Learn more about Tamper Detection.

### 6.9.1.4.4 Loss Of Signal Emit Interval

See Loss Of Signal Emit Interval

## 6.9.2 VCA Deep Learning Skeleton Tracker

The Deep Learning Skeleton tracker tracks people in situations where the camera field of view is relatively close.

The Deep Learning Skeleton Tracker is based on Pose Estimation technology, providing the location of a person in the field of view as well as additional key point metadata on the parts of the body. See Deep Learning Requirements for hardware requirements for this algorithm.

The Deep Learning Skeleton Tracker has the following settings:

### *6.9.2.1 Tamper Detection (DLST)*

Learn more about Tamper Detection.

## 6.9.3 VCA Hand Object Interaction Tracker

The Hand Object Interaction (HOI) Tracker is designed for the detection of hands, and the objects they hold. The HOI tracker requires a top down and relatively close field of view to detect optimally. The list of objects detected by the Hand Object Interaction Tracker is given below:

| Class Name | Description |
|---|---|
| hand | A hand |
| object | An object being held by a hand object |

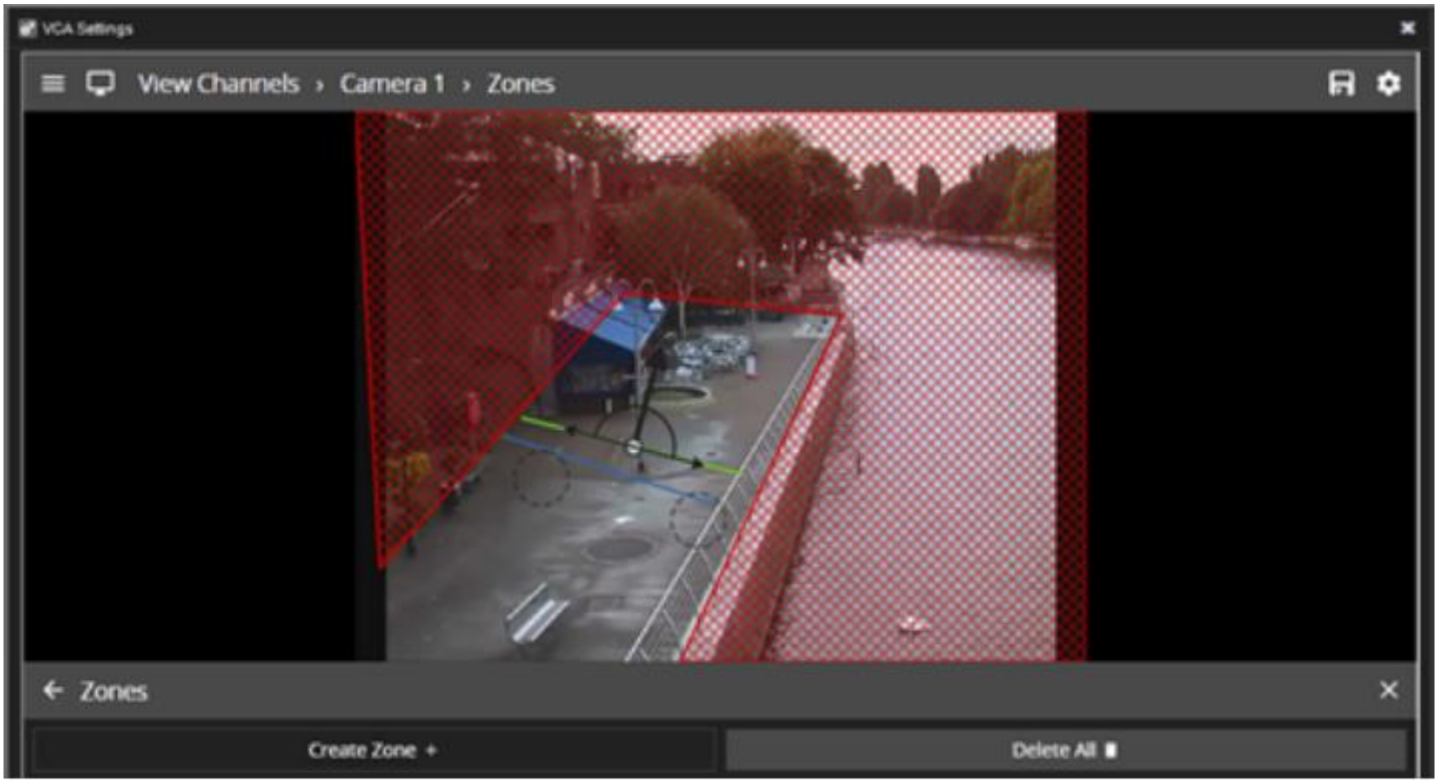
The Hand Object Interaction Tracker is based on a classification and detection model, providing the location of an object in the field of view. See Deep Learning Requirements for hardware requirements for this algorithm.

The Hand Object Interaction Tracker has the following settings:

- Detection Point of Tracked Objects (HOI)
- Tamper Detection (HOI)
- Loss Of Signal Emit Interval (HOI)

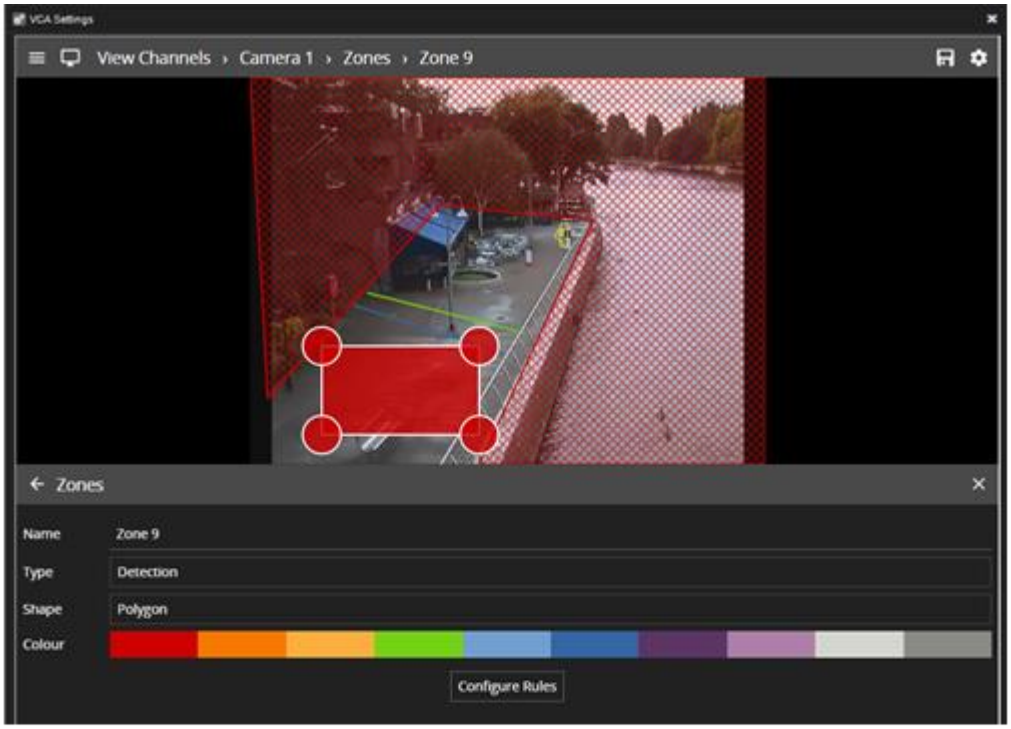
### 6.9.4 VCA Zones

Zones are the detection areas on which VCAcore operate.
To detect a specific behaviour, a zone must be configured to specify the area where a rule applies.



### 6.9.4.1 *Zone specific settings*

The zone configuration menu contains a range of zone-specific configuration parameters:



Name: Zone 9

Type: Detection

Shape: Polygon

- **Name** The name of the zone, which appears in event notifications.
- **Type:** The type of the zone. Can be one of:
- **Detection**: A zone that detects tracked objects and to which rules can be applied.
- **Non-detection:** A zone that specifies an area that should be excluded from VCAcore analysis.
    - Objects are not detected in non-detection zones.
    - Useful for excluding areas of potential nuisance alarms from a scene (e.g. waving trees, flashing lights, etc).
- **Shape**: The shape of the zone. Can be one of:
    - **Polygon:** A polygonal detection area with at least three nodes. Rules apply to the whole area.
    - **Line:** A single- or multi-segment line with at least two nodes. Rules apply to the length of the line.

- **Colour:** The colour of the zone.
- **Configure Rules:** A shortcut button to navigate directly to the rules configuration page

### *6.9.4.2 Adding a zone*

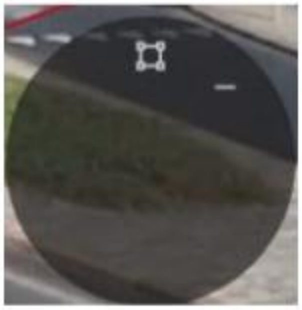
### *6.9.4.3 Zones can be added in multiple ways:*

1. Double-click anywhere on the video display.
2. Click the Create Zone button in the zone settings menu.
3. Right-click or tap-hold to display the context menu and select the add zone icon

⛶

#### 6.9.4.3.1 The context menu

Right-clicking or tap-holding (on mobile devices) displays a context menu that contains commands specific to the current context.



### *6.9.4.3.1.1 The possible actions from the context menu are:*

Adds a new zone.

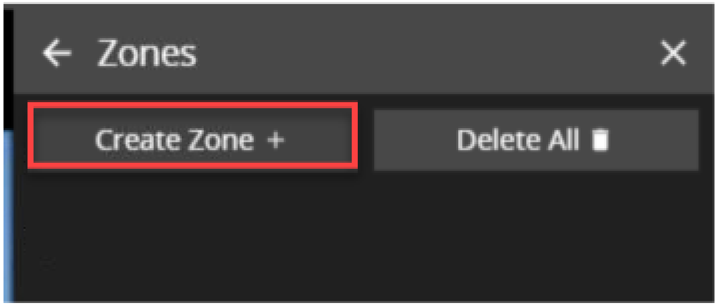
Deletes an existing zone.

Adds a node to a zone.

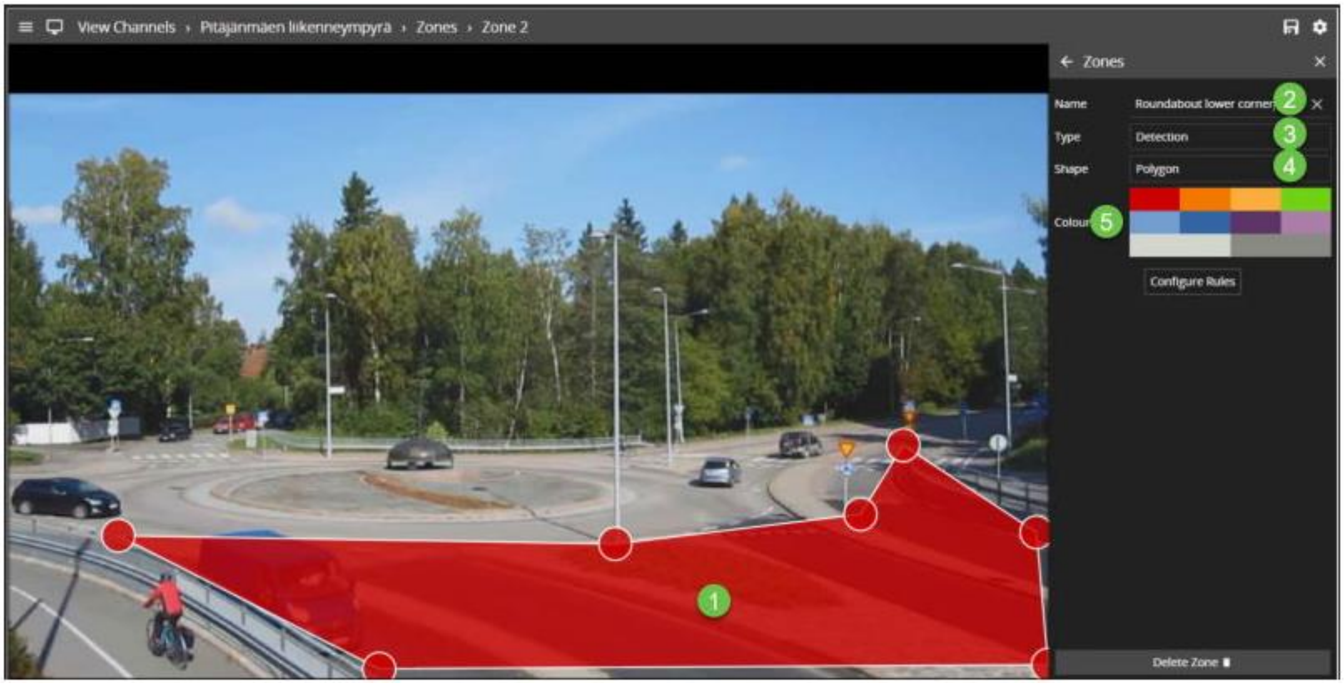
Deletes an existing node from a zone.

### 6.9.4.3.2 Creating a zone

Click **Create Zone** icon



1. Draw zone and set the location to the image
2. Set name of the zone
3. Set detection type
4. Set shape
5. Define color



View Channels › Pitajanmaen liikenneympyrä › Zones › Zone 2

Name: Roundabout lower corner
Type: Detection
Shape: Polygon

### 6.9.4.4 *Positioning zones*

To change the position of a zone, click and drag the zone to a new position. To change the shape of a zone, drag the nodes to create the required shape. New nodes can be added by double-clicking on the edge of the zone or clicking the add node icon  from the context menu.

### 6.9.4.5 *Deleting the zone*

Zones can be deleted in the following ways:

- Select the zone and click the Delete Zone button from the zone settings menu.
- Select the zone, display the context menu and select the delete zone icon

### 6.9.5 VCA Calibration

Camera calibration is required in order for VCAcore to classify objects into different object classes.

Once a channel has been calibrated, VCA Core can infer real-world object properties such as speed, height and area and classify objects accordingly.

Calibration is not needed to do when using Deep Learning tracking, only when using normal VCA or Deep Learning Filter.

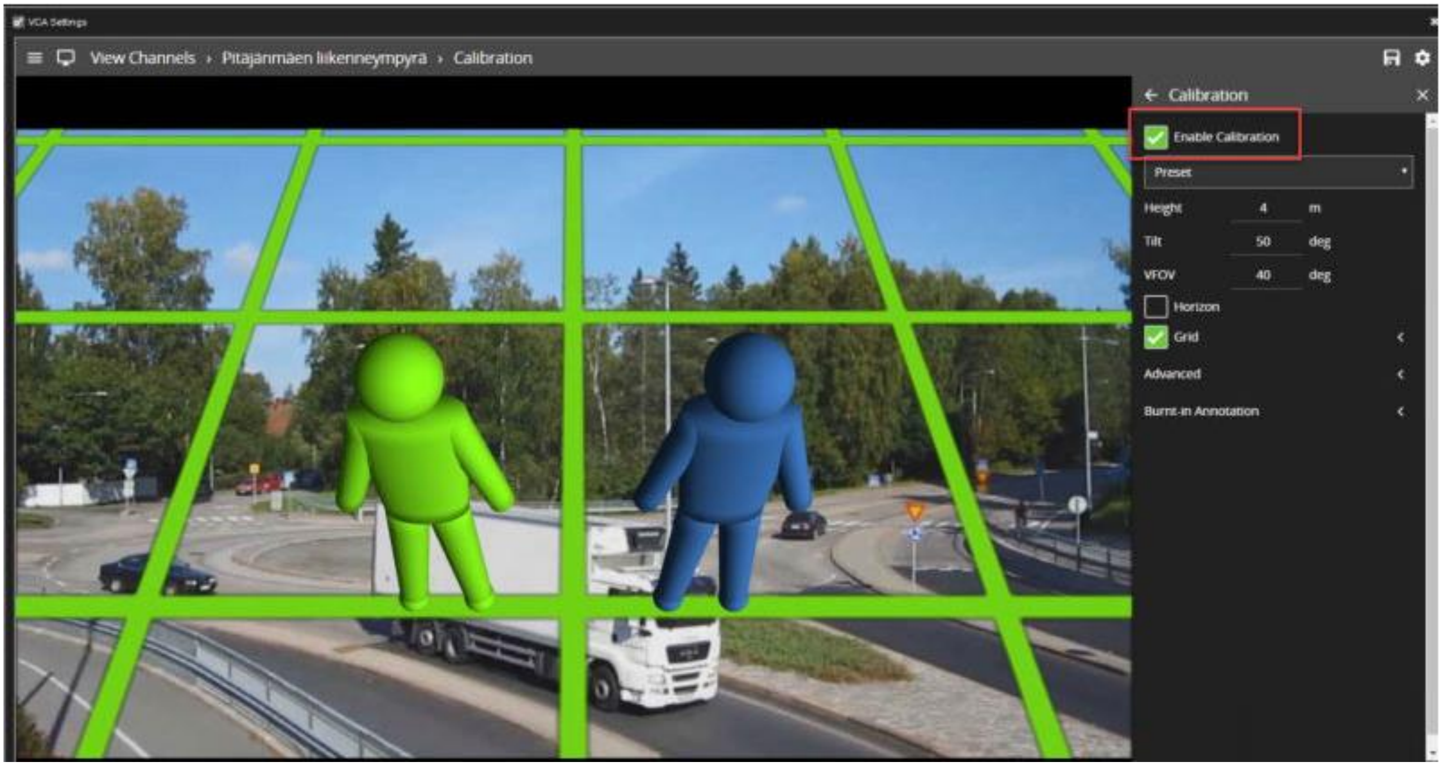
### *6.9.5.1 Camera calibration is split into the following sub-topics:*

1. Enabling Calibration
2. Calibration Controls
3. Calibrating a Channel
4. Advanced Calibration Parameters

### *6.9.5.2 Enabling Calibration*

By default calibration is disabled.

To enable calibration on a channel, check the **Enable Calibration** checkbox.



☑ Enable Calibration

Preset

| | | |
|---|---|---|
| Height | 4 | m |
| Tilt | 50 | deg |
| VFOV | 40 | deg |

□ Horizon

☑ Grid

#### 6.9.5.2.1 Calibration Controls

##### *6.9.5.2.1.1 3D Graphics Overlay*

During the calibration process, the features in the video image need to be matched with a 3D graphics overlay.
The 3D graphics overlay consists of a green grid that represents the ground plane.
Placed on the ground plane are a number of 3D mimics (people-shaped figures) that represent the dimensions of a person with the current calibration parameters.
The calibration mimics are used for verifying the size of a person in the scene and are 1.8 metres tall.
The mimics can be moved around the scene to line up with people (or objects which are of a known, comparable height) to a person.

##### *6.9.5.2.1.2 Mouse Controls*

The calibration parameters can be adjusted with the mouse as follows:

- Click and drag the ground plane to change the camera tilt angle.
- Use the mouse wheel to adjust the camera height. - Drag the slider to change the vertical field of view.

**Note**: The sliders in the control panel can also be used to adjust the camera tilt angle and height.

##### *6.9.5.2.1.3 Control Panel Items*

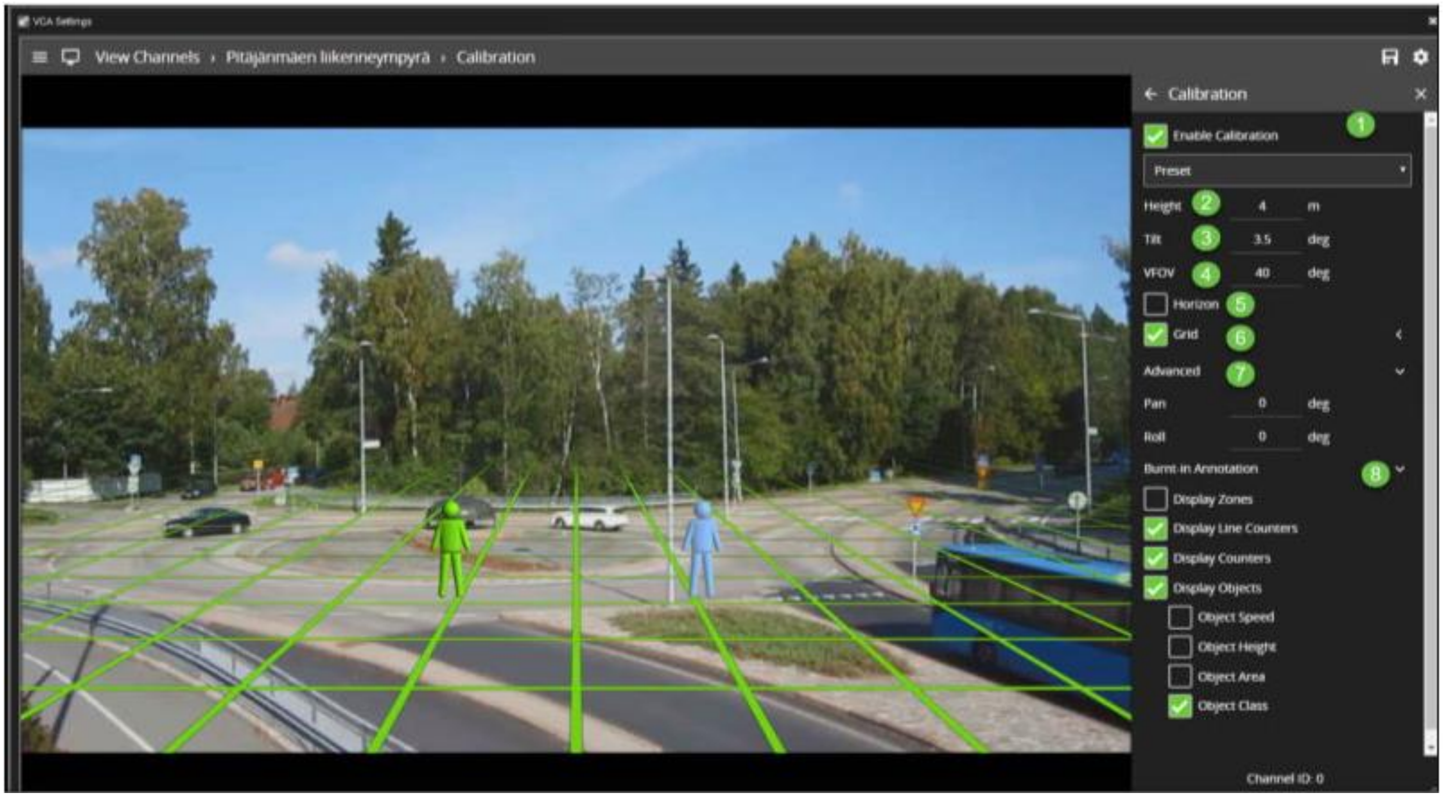
The control panel (shown on the right-hand side in the image above) contains the following controls:

1. **Height**: Adjusts the height of the camera
2. **Tilt**: Adjusts the tilt angle of the camera
3. **VFOV**: Adjusts the vertical field of view of the camera. Note: A correct value for the vertical camera field of view is essential for accurate calibration and classification.
4. **Horizon**: Enables/disables the horizon display. Useful to line up against a horizon in a deep scene.

5. **Grid**: Enables/disables the ground plane grid display. The expand/collapse control (<) exposes additional settings to vary the colour, opacity and size of the ground plane grid.

6. **Advanced**: Exposes advanced settings for controlling the pan and roll of the camera.

7. **Burnt-in Annotation**: Exposes the Burnt-in Annotation controls for convenience.



← Calibration

☑ Enable Calibration

Preset

Height 4 m

Tilt 3.5 deg

VFOV 40 deg

☐ Horizon

☑ Grid

Advanced

Pan 0 deg

Roll 0 deg

Burnt-in Annotation

☐ Display Zones

☑ Display Line Counters

☑ Display Counters

☑ Display Objects

☐ Object Speed

☐ Object Height

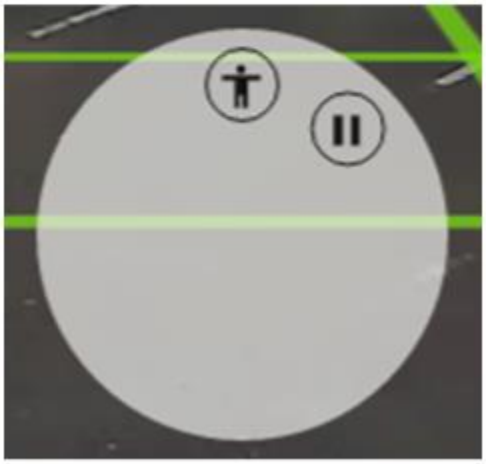
☐ Object Area

☑ Object Class

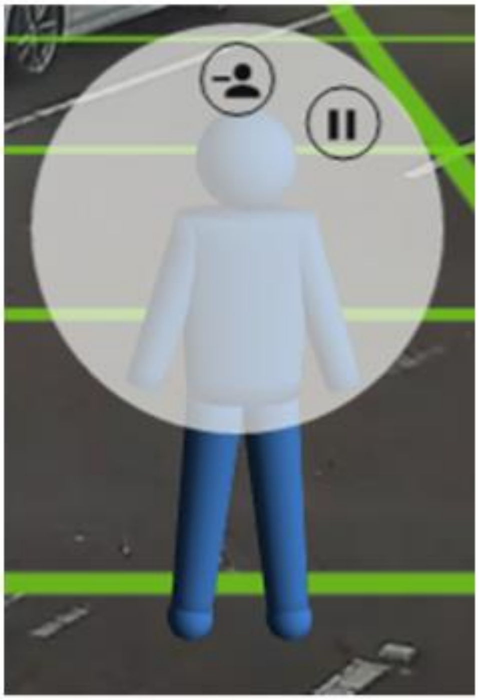
Channel ID: 0

### 6.9.5.2.1.4 Context Menu Items

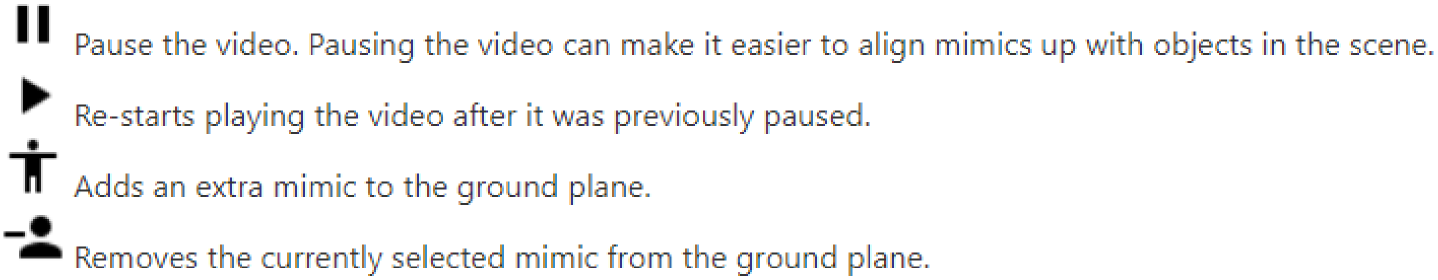
Right-clicking the mouse (or tap-and-hold on a tablet) on the grid displays the context menu:



Performing the same action on a mimic display the mimic context menu:



The possible actions from the context menu are:



⏸ Pause the video. Pausing the video can make it easier to align mimics up with objects in the scene.

▶ Re-starts playing the video after it was previously paused.

🧍 Adds an extra mimic to the ground plane.

−👤 Removes the currently selected mimic from the ground plane.

### *6.9.5.3 Calibrating a Channel*

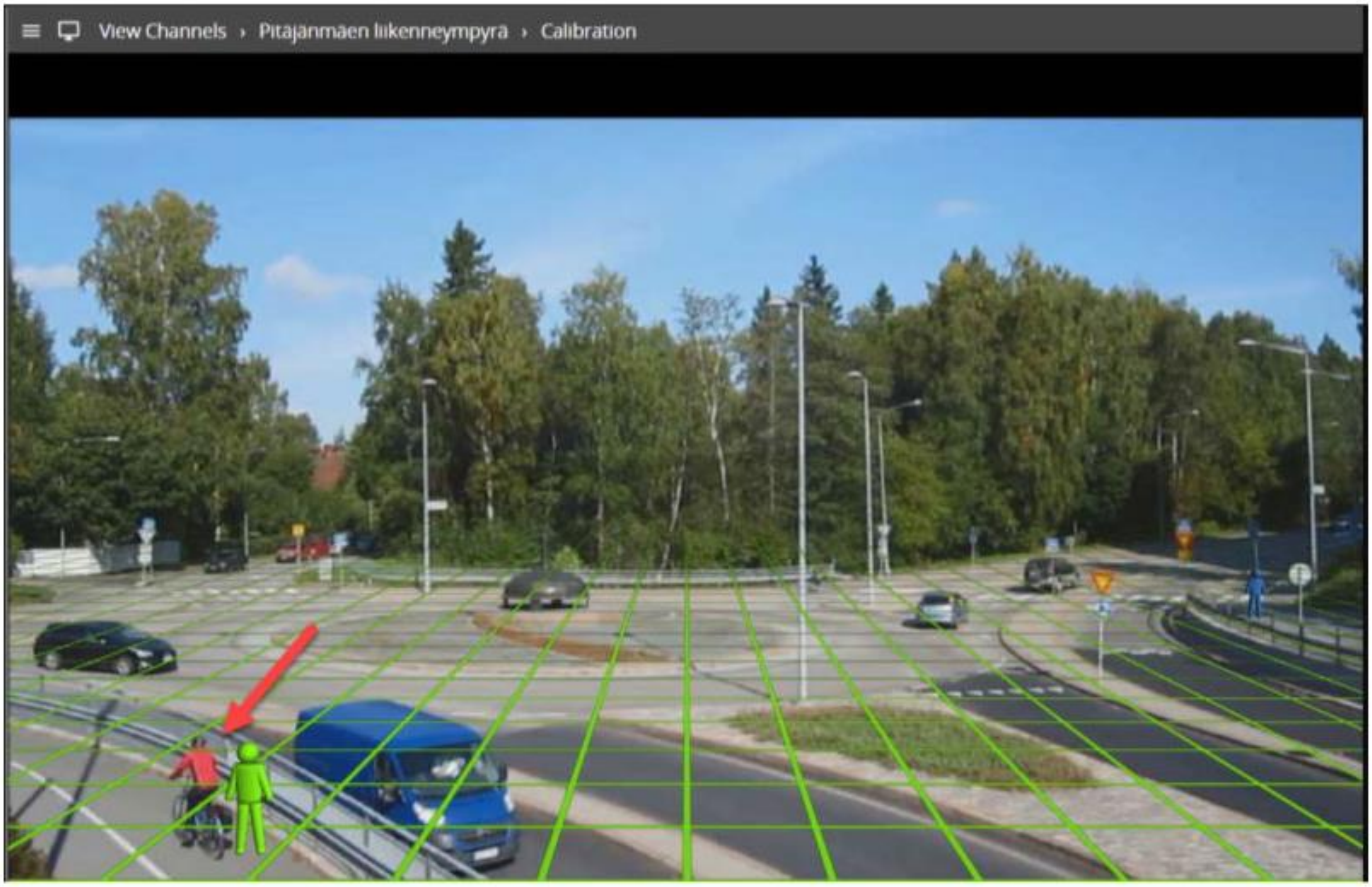
Calibrating a channel is necessary in order to estimate object parameters such as height, area, speed and classification.

If the height, tilt angle and vertical field of view corresponding to the installation are known, these can simply be entered as parameters in the appropriate fields in the control panel.

If however, these parameters are not explicitly known this section provides a step-by-step guide to calibrating a channel.

#### 6.9.5.3.1 Step 1: Find People in the Scene

Find some people or some people-sized objects in the scene.
Try to find a person near the camera, and a person further away from the camera.
It is useful to use the play/pause control to pause the video so that the mimics can be accurately placed. Place the mimics on top of or near the people:



### 6.9.5.3.2 Step 2: Enter the Camera Vertical Field of View

Determining the correct vertical field of view is important for accurate calibration.

The following table shows pre-calculated values for the vertical field of view for different sensor sizes.

| | Focal Length(mm) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 15 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCD Size (in) | CCD Height(mm) | | | | | | | | | | | | | | | |
| 1/6" | 1.73 | 82 | 47 | 32 | 24 | 20 | 16 | 14 | 12 | 11 | 10 | 7 | | | | |
| 1/4" | 2.40 | 100 | 62 | 44 | 33 | 27 | 23 | 19 | 17 | 15 | 14 | 9 | 7 | | | |
| 1/3.6" | 3.00 | 113 | 74 | 53 | 41 | 33 | 28 | 24 | 21 | 19 | 12 | 11 | 9 | 6 | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1/3.2" | 3.42 | 119 | 81 | 59 | 46 | 38 | 32 | 27 | 24 | 21 | 16 | 13 | 10 | 7 | | |
| 1/3" | 3.60 | 122 | 84 | 62 | 48 | 40 | 33 | 29 | 25 | 23 | 20 | 14 | 10 | 7 | 5 | |
| 1/2.7" | 3.96 | 126 | 89 | 67 | 53 | 43 | 37 | 32 | 28 | 25 | 22 | 15 | 11 | 8 | 6 | |
| 1/2" | 4.80 | 135 | 100 | 77 | 62 | 51 | 44 | 38 | 33 | 30 | 27 | 18 | 14 | 9 | 7 | 5 |
| 1/1.8" | 5.32 | 139 | 106 | 83 | 67 | 56 | 48 | 42 | 37 | 33 | 30 | 20 | 15 | 10 | 8 | 6 |
| 2/3" | 6.60 | | 118 | 95 | 79 | 67 | 58 | 50 | 45 | 40 | 37 | 25 | 19 | 13 | 9 | 8 |
| 1" | 9.60 | | 135 | 116 | 100 | 88 | 77 | 69 | 62 | 56 | 51 | 35 | 27 | 18 | 14 | 11 |
| 4/3" | 13.50 | | | 132 | 119 | 107 | 97 | 88 | 80 | 74 | 68 | 48 | 37 | 25 | 19 | 15 |

If the table does not contain the relevant parameters, the vertical FOV can be estimated by viewing the extremes of the image at the top and bottom.

Note that without the correct vertical FOV, it may not be possible to get the mimics to match people at different positions in the scene.

#### 6.9.5.3.3 Step 3: Enter the Camera Height

If the camera height is known, type it indirectly. If the height is not known, estimate it as far as possible and type it indirectly.

#### 6.9.5.3.4 Step 4: Adjust the Tilt Angle and Camera Height

Adjust the camera tilt angle (and height if necessary) until both mimics are approximately the same size as a real person at that position in the scene.

Click and drag the ground plane to change the tilt angle and use the mouse wheel or control panel to adjust the camera height.

The objective is to ensure that mimics placed at various locations on the grid line up with people or people-sized- objects in the scene.

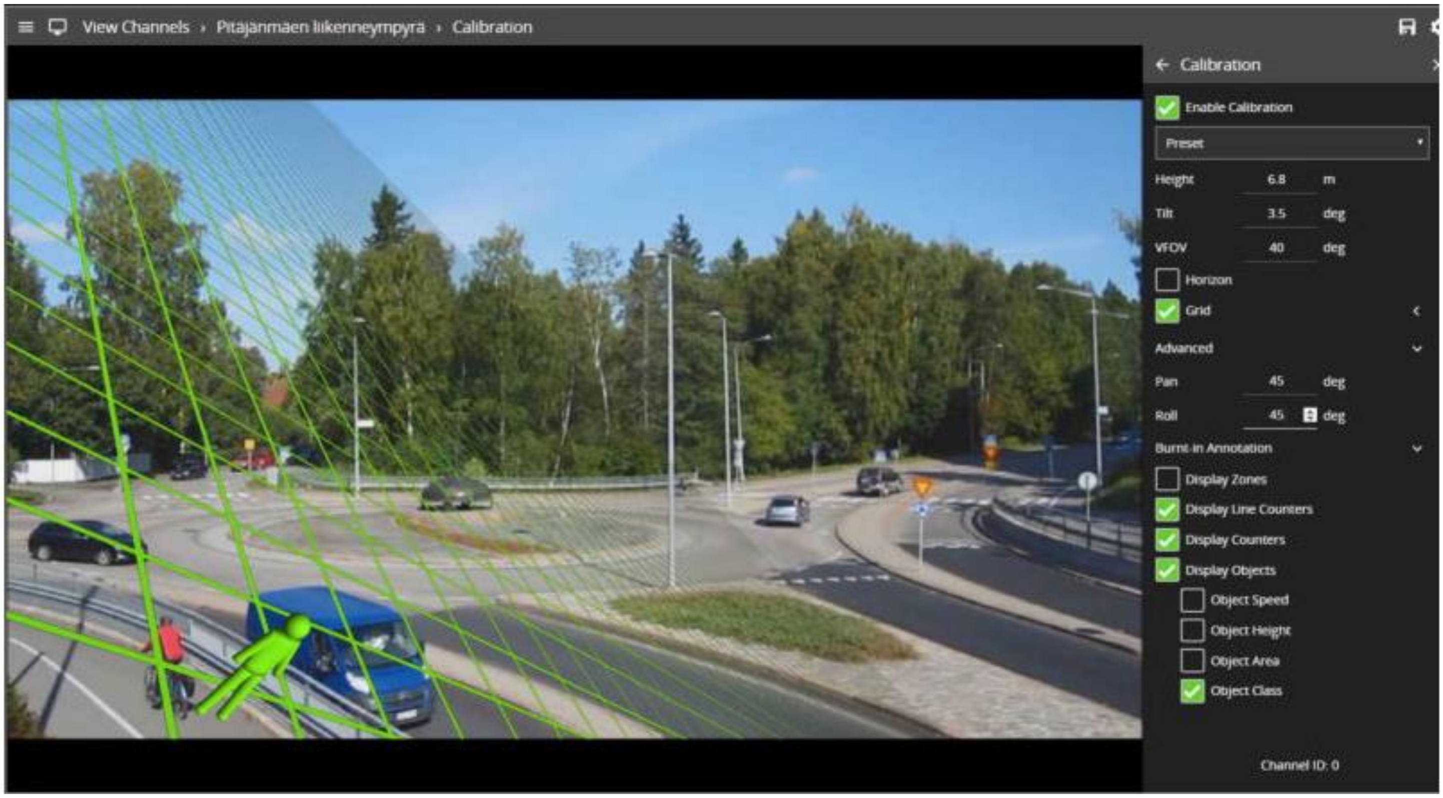
Once the parameters have been adjusted, the object annotation will reflect the changes and classify the objects accordingly.

#### 6.9.5.3.5 Step 5: Verify the Setup

- Once the scene is calibrated, drag or add mimics to different locations in the scene and verify they appear at the same size/height as a real person would.
- Validate that the height and area reported by the VCAcore annotation look approximately correct.
- Note that the burnt-in -annotation settings in the control panel can be used to enable and disable the different types of annotation.
- Repeat step 4 until the calibration is acceptable.

### *6.9.5.4 Advanced Calibration Parameters*

The advanced calibration parameters allow the ground plane to be panned and rolled without affecting the camera calibration parameters.
This can be useful to visualize the calibration setup if the scene has a pan or roll with respect to the camera.



☑ Enable Calibration

Preset

Height 6.8 m

Tilt 3.5 deg

VFOV 40 deg

☐ Horizon

☑ Grid

Advanced

Pan 45 deg

Roll 45 deg

Burnt-in Annotation

☐ Display Zones

☑ Display Line Counters

☑ Display Counters

☑ Display Objects

☐ Object Speed

☐ Object Height

☐ Object Area

☑ Object Class

Channel ID: 0

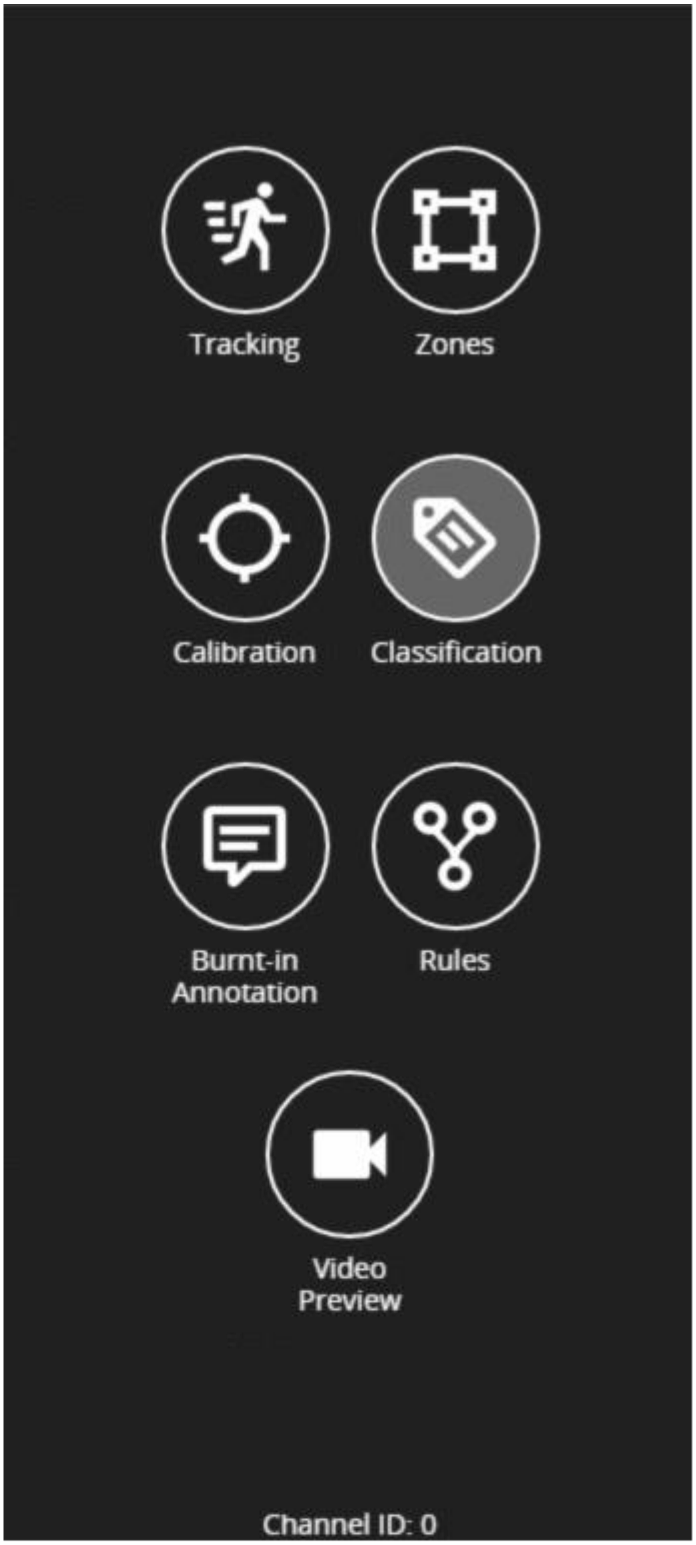
**Note:** the pan and roll advanced parameters only affect the orientation of the 3D ground plane so that it can be more conveniently aligned with the video scene, and does not actually affect the calibration parameters.

### 6.9.6 VCA Classification

VCAcore can define a moving objects class using either its Deep Learning models or by using properties extracted from an object in a calibrated scene.

Both methods of classification are applied through the use of filters in the rules interface.

Classification filters allow an object, which has triggered a rule, to be evaluated against its predicted class and filtered out if needed.


Tracking
Zones
Calibration
Classification
Burnt-in Annotation
Rules
Video Preview
Channel ID: 0

#### *6.9.6.1 Object classification*

Once a camera view has been calibrated, each detected object in that view will have a number of properties extracted including object area and speed.

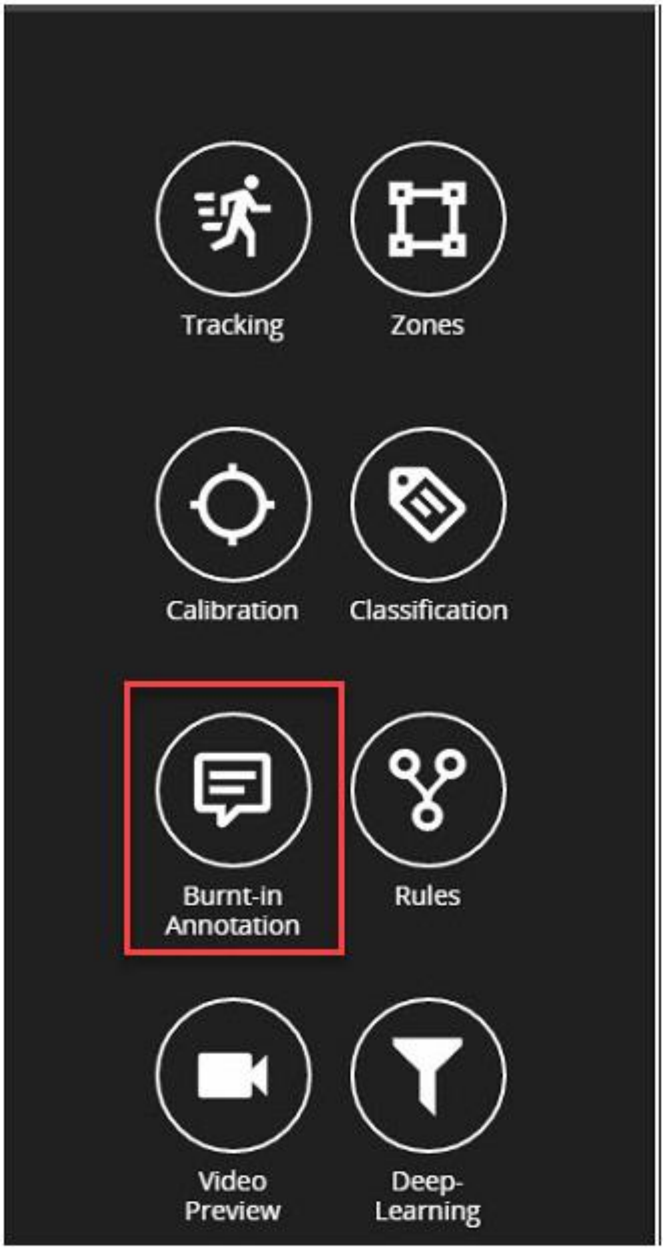
VCAserver's object classification performs classification by comparing these properties to a set of configurable object classifiers.

VCAserver comes pre-loaded with the most common object classifiers, and in most cases these will not need to be modified.

### 6.9.7 VCA Burnt-in Annotation

Burnt-in Annotations allow VCAserver metadata to be overlaid on to the raw video stream.

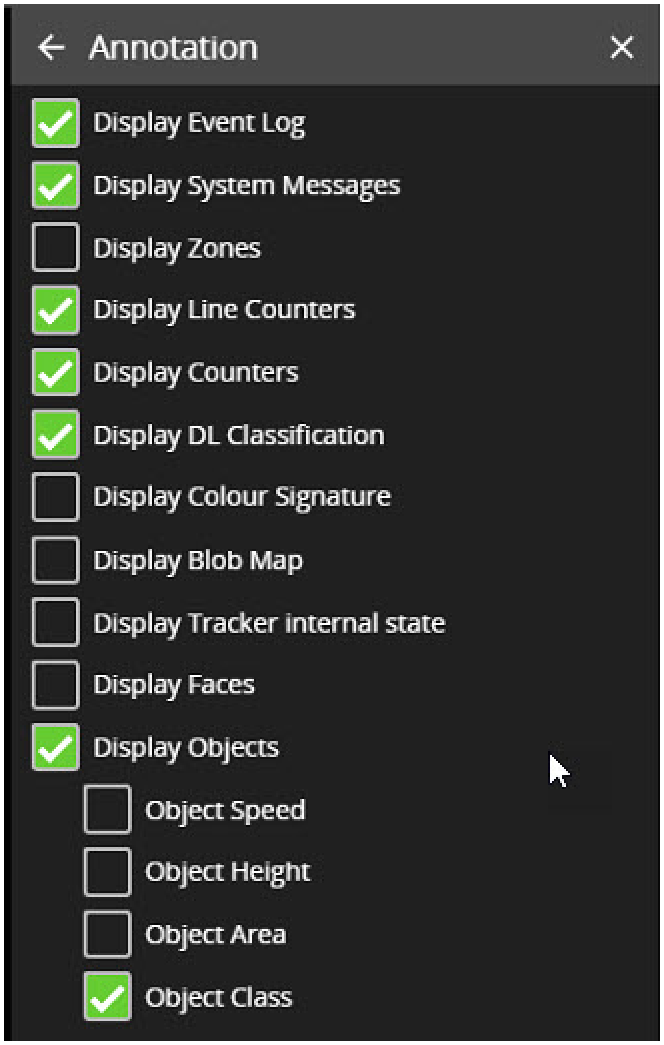
The burnt-in annotation settings control which VCAserver metadata (objects, events, etc) is rendered into the video stream.



**Note:**

- To display object parameters such as speed, height, area and classifications, the channel must first be calibrated.

- To display DL Classification annotations, the channel must have an active Deep Learning Filter rule configured or the DL People Tracker enabled.
- To display colour signature annotations, the channel must have an active Colour Filter rule configured.
- Some annotations only apply to certain trackers, in such cases the required tracker is listed in brackets.



← Annotation ✕

- ☑ Display Event Log
- ☑ Display System Messages
- □ Display Zones
- ☑ Display Line Counters
- ☑ Display Counters
- ☑ Display DL Classification
- □ Display Colour Signature
- □ Display Blob Map
- □ Display Tracker internal state
- □ Display Faces
- ☑ Display Objects
  - □ Object Speed
  - □ Object Height
  - □ Object Area
  - ☑ Object Class

### 6.9.8 VCA Rules

VCAcore's rules are used to detect specific events in a video stream.
There are three rule types that can be utilized to detect events and trigger actions:

#### *6.9.8.1 Basic Inputs / Rule:*

An algorithm that will trigger when a particular behaviour or event has been observed e.g. Presence. Basic inputs can be used to trigger an action.

#### *6.9.8.2 Filters:*

A filter that will trigger if the object which has triggered the input rule / logical rule meets the filter requirements e.g. is moving at a specific speed. Filters can be used to trigger an action.

#### *6.9.8.3 Conditional Rule:*

A logical link between one or more inputs to allow the detection of more complex behaviours e.g. AND.
Conditional rules can be used to trigger an action.
Within VCAcore, rule configurations can be as simple as individual basic inputs attached to a zone used to trigger an action.

Alternatively, rules can be combined into more complex logical rule configurations using conditional rules and filters.

The overarching goal of the rules in VCAcore is to help eliminate erroneous alerts being generated by providing functions to prevent unwanted behaviour from triggering an action.

- VCA - Basic inputs
- VCA - Aggressive Behaviour
- VCA - Abandoned and Removed object
- VCA - Appear and Disappear
- VCA - Enter and exit
- VCA - Direction
- VCA - Directional Crossing

- VCA - Dwell
- VCA - Fall
- VCA - Presence
- VCA - Tailgating
- VCA - Stopped
- VCA - Counting Line
- VCA - Conditional rules
- VCA - Counters
- VCA - Object trails
- VCA - Object display

### *6.9.8.4 VCA - Basic inputs*

A basic input or rule can only be used to trigger an action or as an input to another rule type. Basic inputs always require a zone, and potentially some additional parameters.
A basic input can be used on its own to trigger an action, although they are often used as an input to other filters or conditional rules.

### *6.9.8.5 VCA - Aggressive Behaviour*

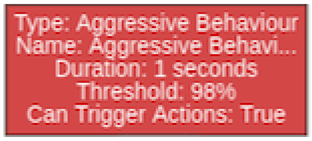
A rule which fires when aggressive behaviour is detected in the field of view for longer than the specified duration.

> Aggressive behaviour does not require a zone and runs independently of the tracker. Enabling this algorithm, by adding this rule, will impact channel capacity, as the algorithm runs in addition to the channels selected tracker.

> Separate VCAbehaviour license needed for this feature.

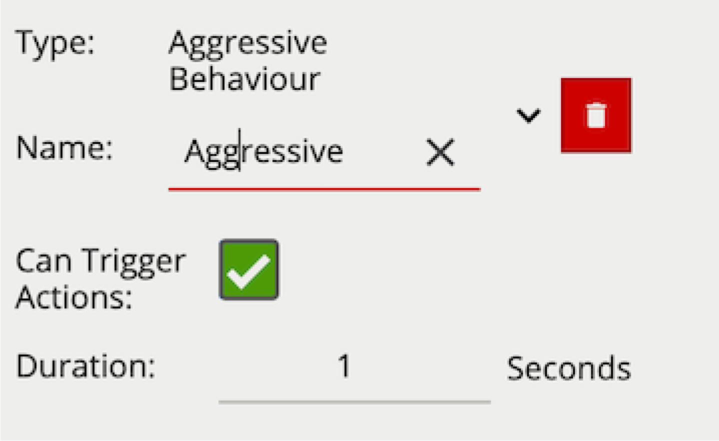
100294 | 2023-03-29 11:58:01 | Agressive | Aggressive Behaviour

#### 6.9.8.5.1 Graphical View



Type: Aggressive Behaviour
Name: Aggressive Behavi...
Duration: 1 seconds
Threshold: 98%
Can Trigger Actions: True

#### 6.9.8.5.2 Form View



Type: Aggressive Behaviour

Name: Aggressive

Can Trigger Actions: ☑

Duration: 1 Seconds

### 6.9.8.5.3 Configuration

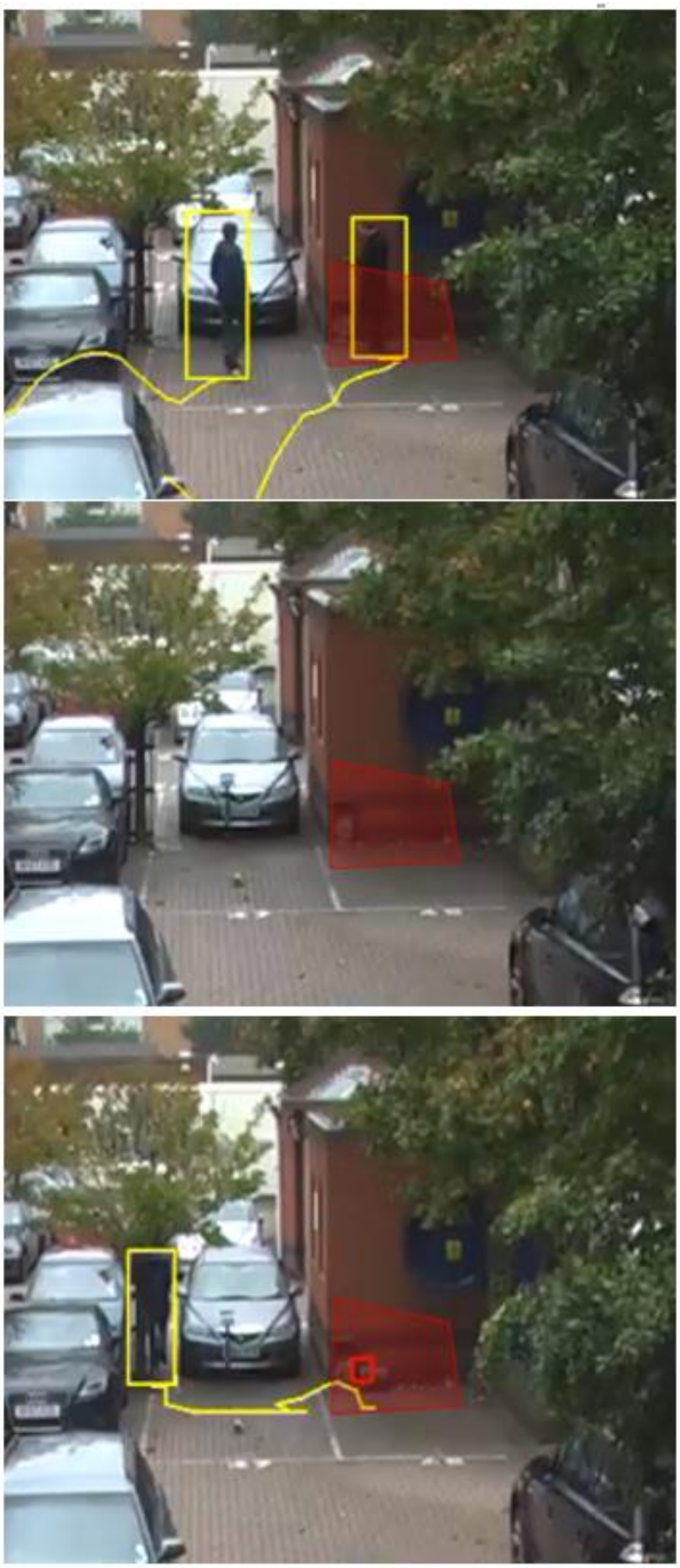
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Aggressive #” |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Duration | Period of time before an aggressive behaviour triggers the rule | 1 to 60 seconds |

### *6.9.8.6 VCA - Abandoned and Removed object*

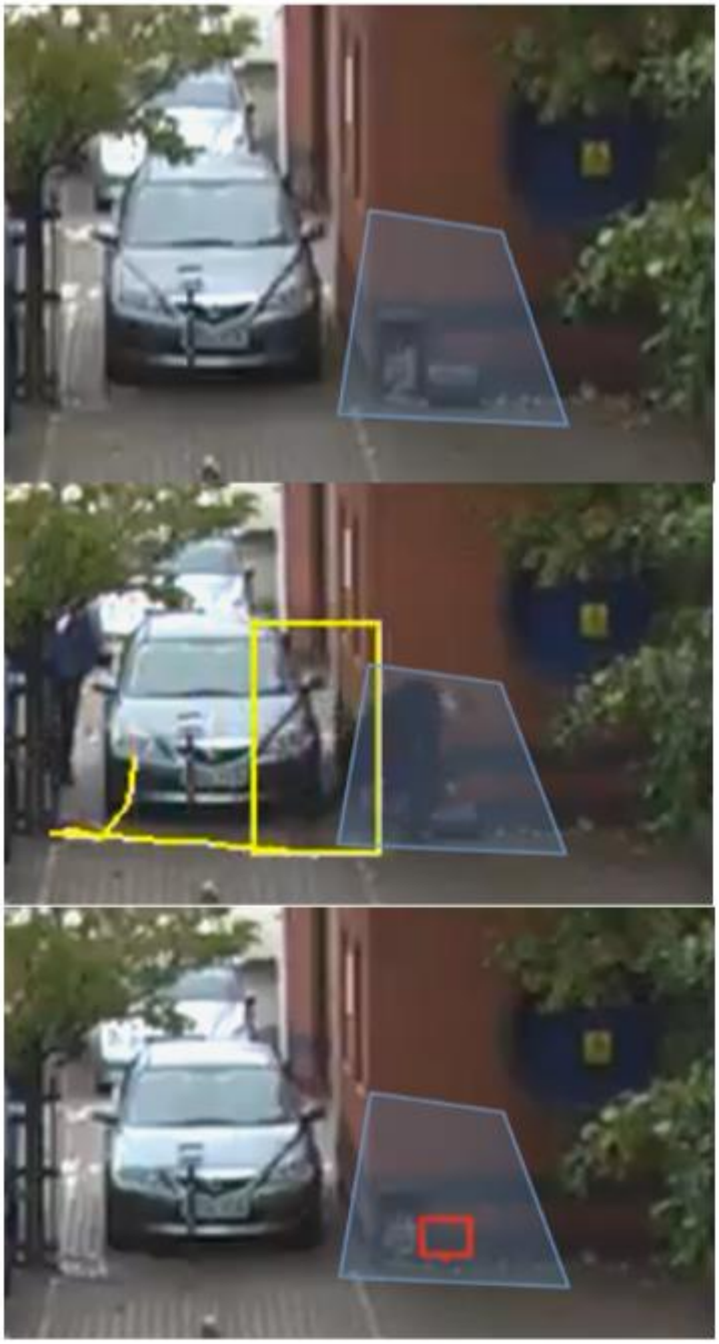
The abandoned and removed object rule triggers when an object has been either left within a defined zone, e.g. a person leaving a bag on a train platform, or when an object is removed from a defined zone.

The abandoned rule has a duration property which defines the amount of time an object must have been abandoned for or removed, to trigger the rule.

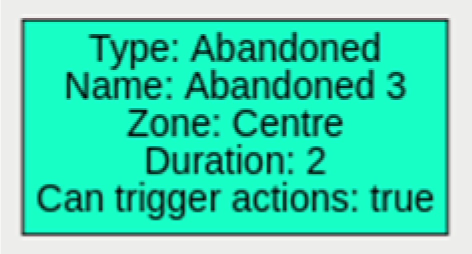
**Below is a sample scenario where a bag is left in a defined zone resulting in the rule triggering.**



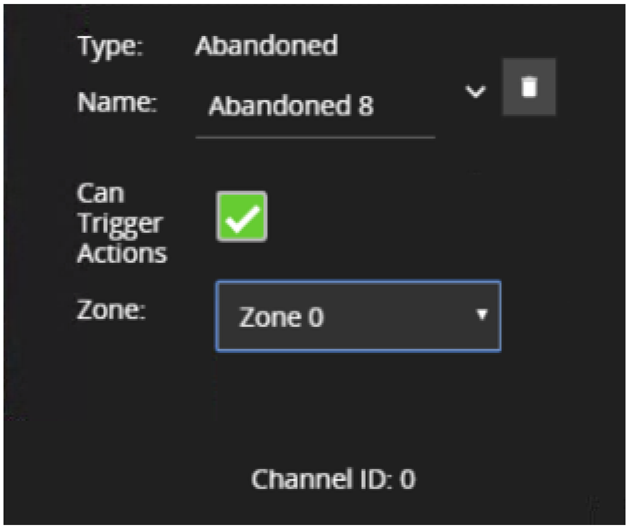
**Below is a similar example scenario where the bag is removed from the defined zone resulting in the rule triggering.**



**Note:** The algorithm used for abandoned and removed object detection is the same in each case, and therefore cannot differentiate between objects which have been abandoned or removed. This arises because the algorithm only analyses how blocks of pixels change with respect to a background model which is constructed over time.



Type: Abandoned
Name: Abandoned 3
Zone: Centre
Duration: 2
Can trigger actions: true



Type: Abandoned
Name: Abandoned 8
Can Trigger Actions ☑
Zone: Zone 0
Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Abandoned #" |
| Zone | The zone this rule is associated with | None |
| Duration | Period of time an object must have been abandoned or removed before the rule triggers | 0 |

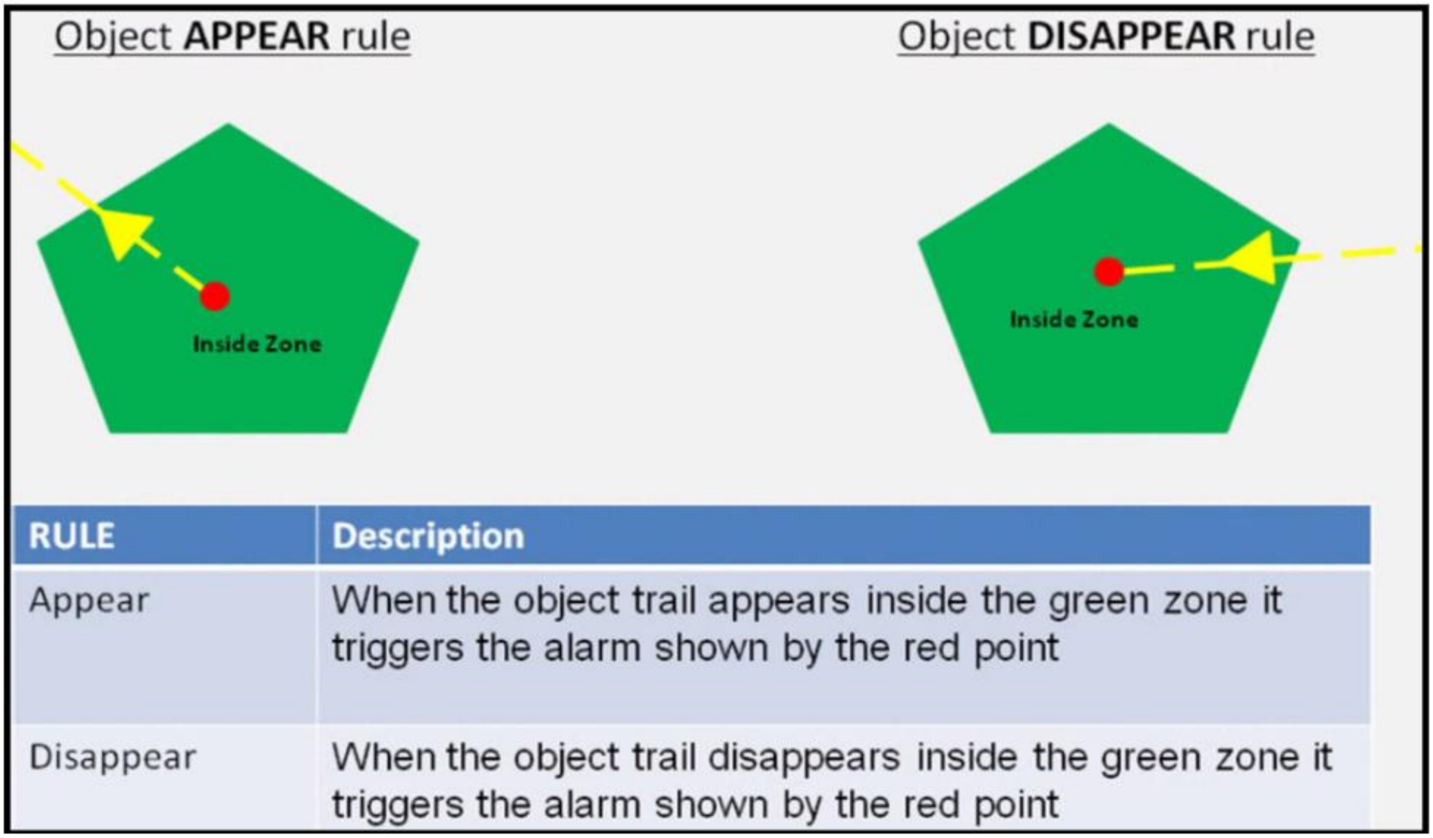
| | | |
|---|---|---|
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

### *6.9.8.7 VCA - Appear and Disappear*

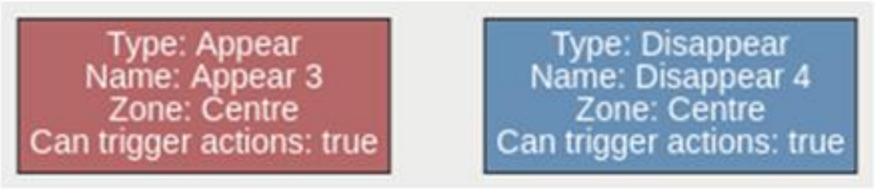
The appear rule detects objects that start being tracked within a zone, e.g. a person who appears in the scene from a doorway.

Conversely, the disappear rule detects objects that stop being tracked within a zone, e.g. a person who exits the scene through a doorway.

Note: The appear and disappear rules differ from the enter and exit rules as detailed in the enter and exit rule descriptions.

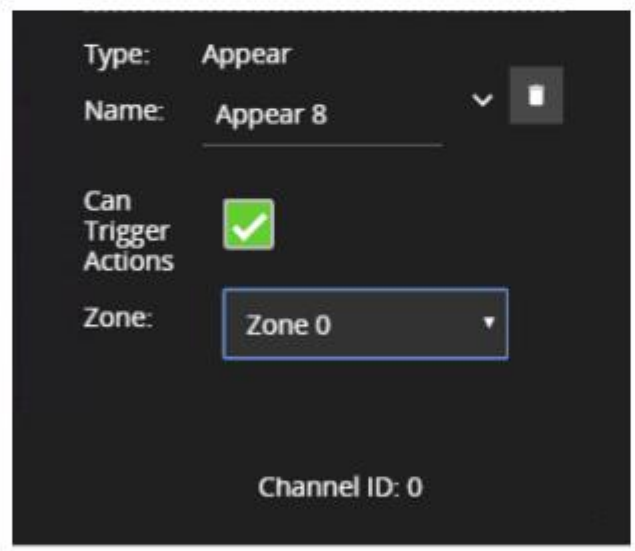


| RULE | Description |
|---|---|
| Appear | When the object trail appears inside the green zone it triggers the alarm shown by the red point |
| Disappear | When the object trail disappears inside the green zone it triggers the alarm shown by the red point |



Type: Appear
Name: Appear 3
Zone: Centre
Can trigger actions: true

Type: Disappear
Name: Disappear 4
Zone: Centre
Can trigger actions: true



Type: Appear

Name: Appear 8

Can Trigger Actions ☑

Zone: Zone 0

Channel ID: 0

### 6.9.8.7.1 Configuration Appear

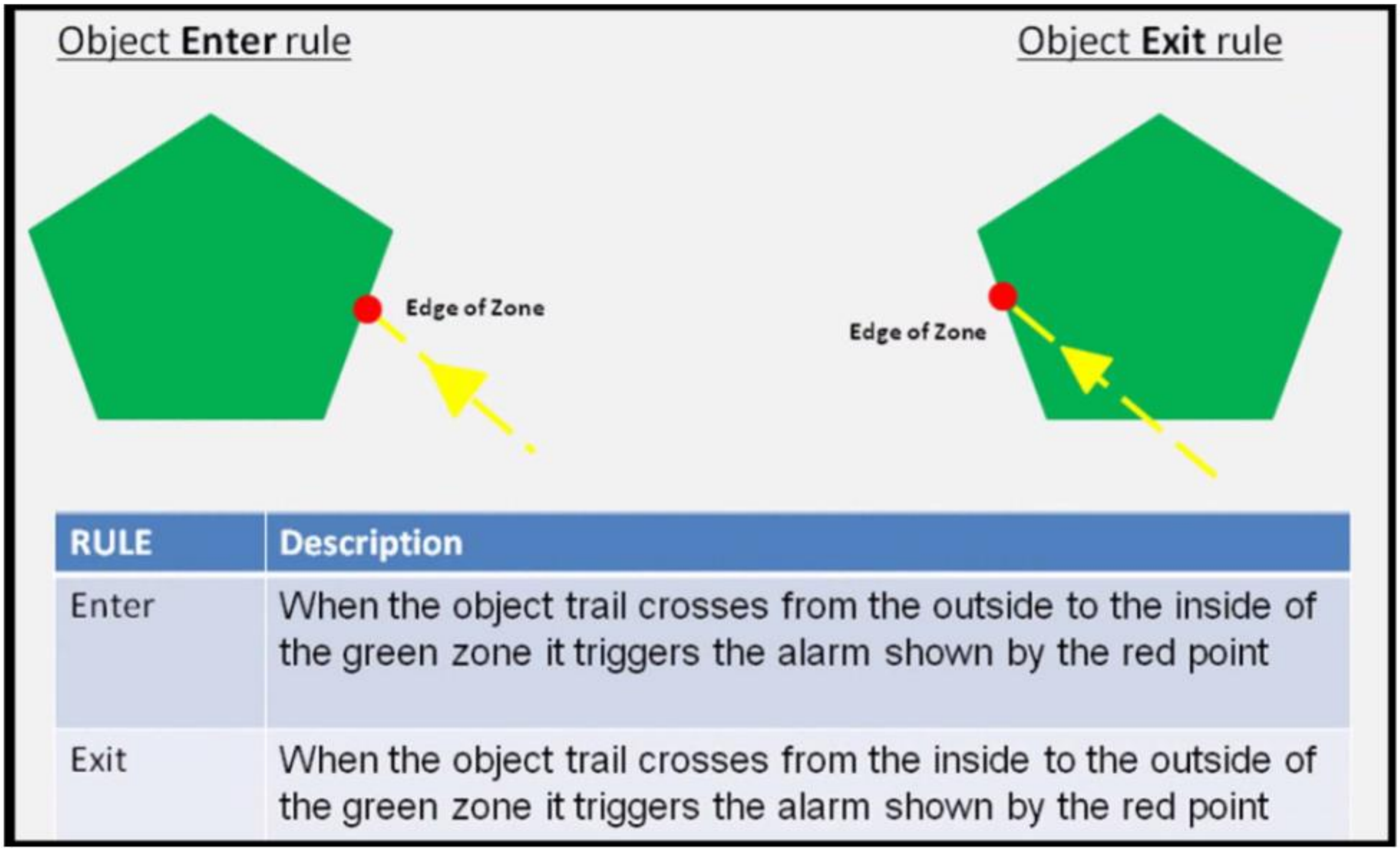
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Appear #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |

### 6.9.8.7.2 Configuration Disappear

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Disappear #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

| | | |
|---|---|---|
| Zone | The zone this rule is associated with | None |

### 6.9.8.8 VCA - Enter and exit



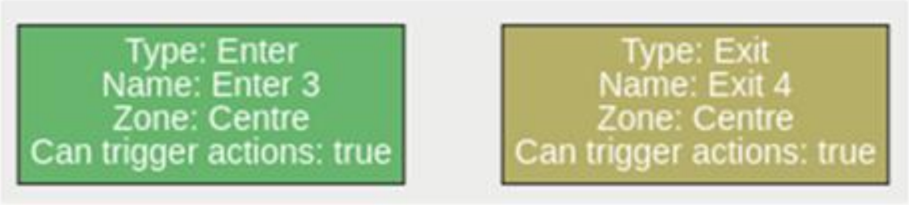
| RULE | Description |
|---|---|
| Enter | When the object trail crosses from the outside to the inside of the green zone it triggers the alarm shown by the red point |
| Exit | When the object trail crosses from the inside to the outside of the green zone it triggers the alarm shown by the red point |

The enter rule detects when objects enter a zone.
In other words, when objects cross from the outside of a zone to the inside of a zone.
Conversely, the exit rule detects when an object leaves a zone: when it crosses the border of a zone from the inside to the outside.

**Note:** Enter and exit rules differ from appear and disappear rules, as follows:

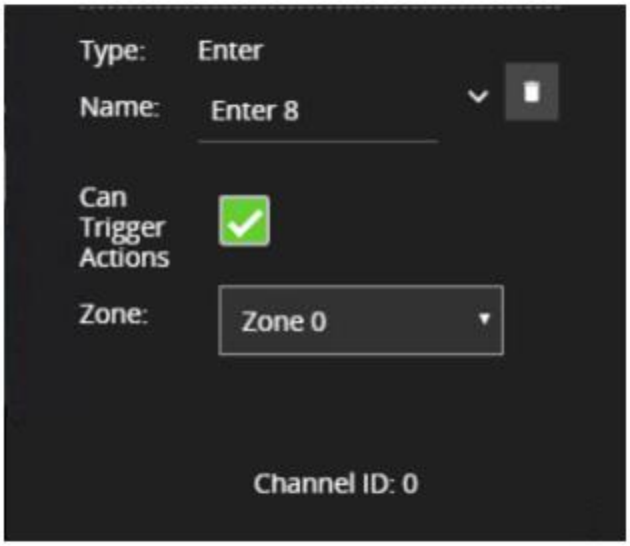
1. Whereas the enter rule detects already-tracked objects crossing the zone border from outside to inside, the appear rule detects objects which start being tracked within a zone (e.g. appear in the scene through a door).

2. Whereas the exit rule detects already-tracked objects crossing the zone border from inside to outside, the disappear rule detects objects which stop being tracked within the zone (e.g. leave the scene through a door).

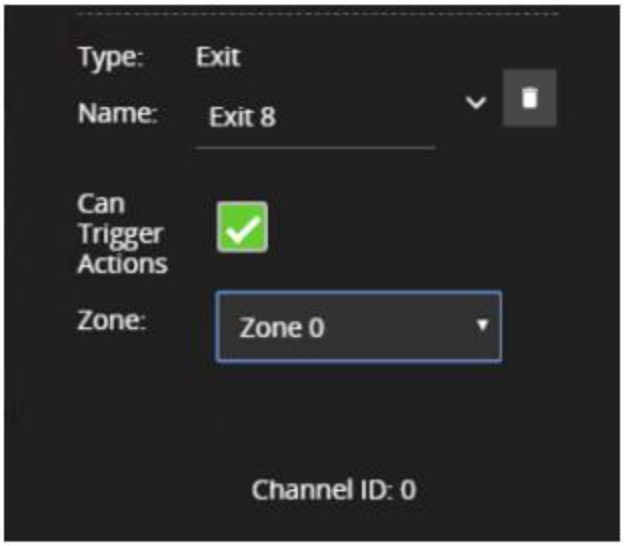


Type: Enter
Name: Enter 3
Zone: Centre
Can trigger actions: true

Type: Exit
Name: Exit 4
Zone: Centre
Can trigger actions: true



Type: Enter
Name: Enter 8
Can Trigger Actions ☑
Zone: Zone 0
Channel ID: 0



Type: Exit
Name: Exit 8
Can Trigger Actions ☑
Zone: Zone 0
Channel ID: 0

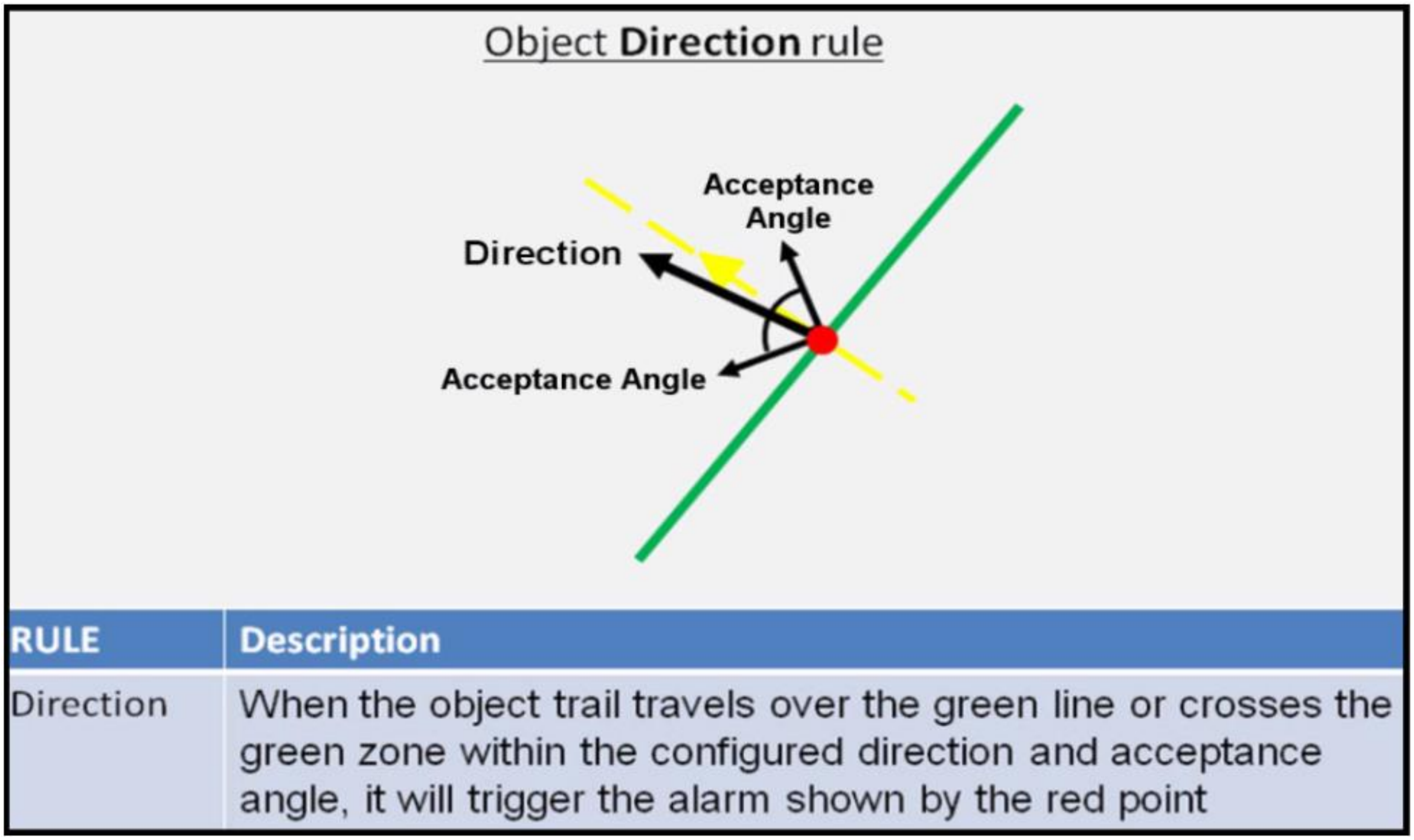
#### 6.9.8.8.1 Configuration Enter

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Enter #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |

#### 6.9.8.8.2 Configuration Exit

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Exit #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |

### *6.9.8.9 VCA - Direction*



Object **Direction** rule

| RULE | Description |
|---|---|
| Direction | When the object trail travels over the green line or crosses the green zone within the configured direction and acceptance angle, it will trigger the alarm shown by the red point |

The direction rule detects objects moving in a specific direction.

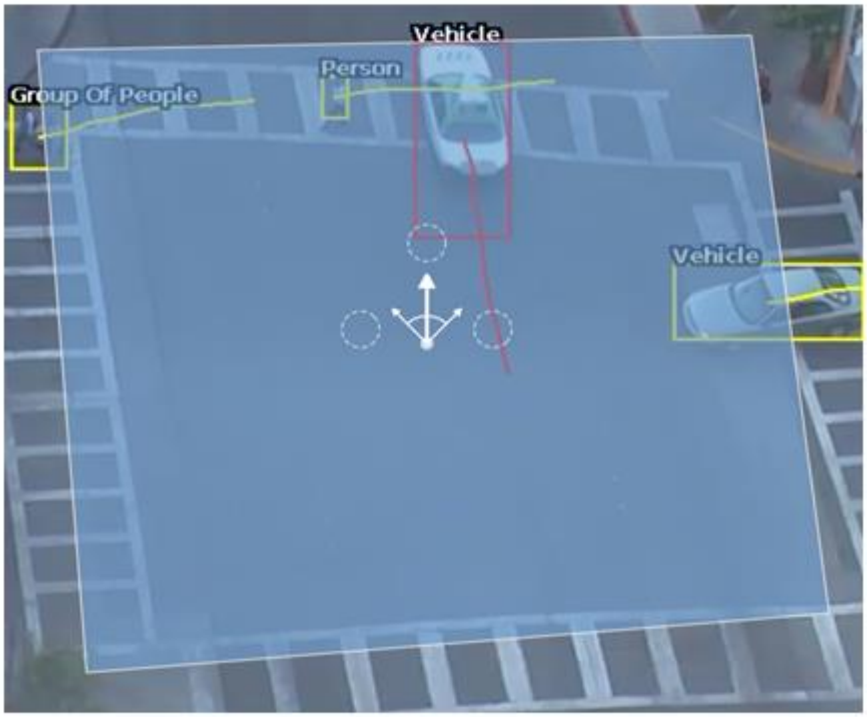
Configure the direction and acceptance angle by moving the arrows on the direction control widget.

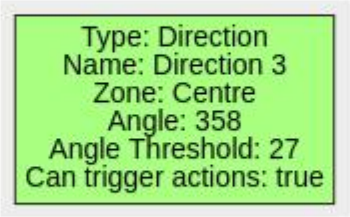
The primary direction is indicated by the large central arrow.

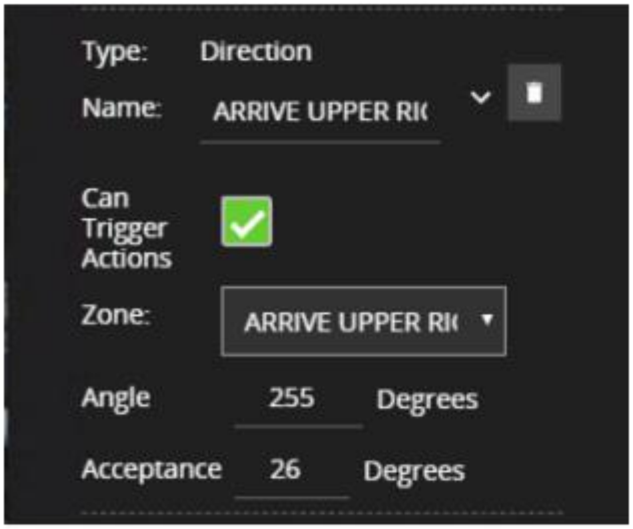
The acceptance angle is the angle between the two smaller arrows.

Objects that travel in the configured direction (within the limits of the acceptance angle), through a zone or over a line, trigger the rule and raise an event.

The following image illustrates how the white car moving in the configured direction triggers the rule whereas the other objects do not.




Type: Direction
Name: Direction 3
Zone: Centre
Angle: 358
Angle Threshold: 27
Can trigger actions: true


Type: Direction

Name: ARRIVE UPPER RI(

Can Trigger Actions ☑

Zone: ARRIVE UPPER RI( ▾

Angle 255 Degrees

Acceptance 26 Degrees

### 6.9.8.9.1 Configuration Direction

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Direction #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |
| Angle | Primary direction angle, 0 - 359. 0 references up. | 0 |
| Acceptance | Allowed variance on each side of the primary direction that will still trigger rule | 0 |

### *6.9.8.10 VCA - Directional Crossing*

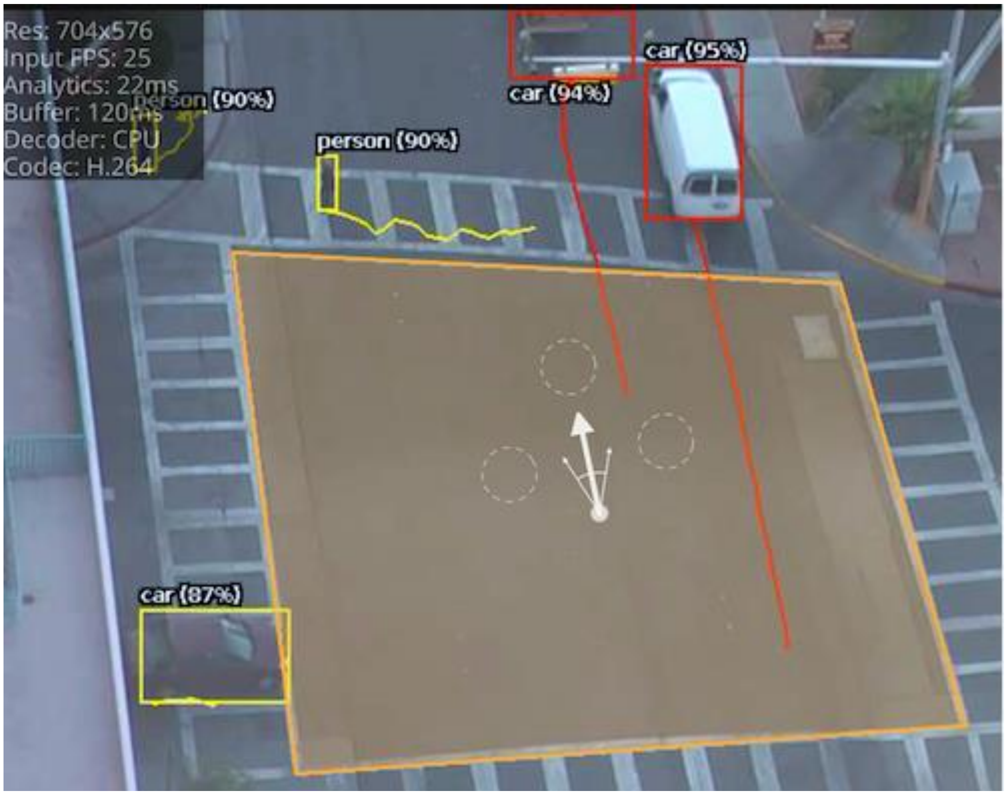
The directional crossing rule is designed to reduce false alarms common with simple line crossing use cases. Directional Crossing is designed for use with a zone rather than a line, and adds a number of additional checks for an object as it enters as well as exits that zone.

For an object to trigger the Directional Crossing rule it must:

- Enter the zone travelling in a direction that falls within the acceptance angle.
- Be classified as one of the specified object classes.
- Exit that zone travelling in a direction that falls within the acceptance angle.

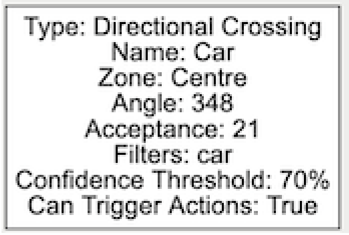
Configure the direction and acceptance angle by moving the arrows on the direction control widget. The primary direction is indicated by the large central arrow. The acceptance angle is the angle between the two smaller arrows.

The following image illustrates how the white car, moving in the configured direction, triggers the rule whereas the other objects do not.



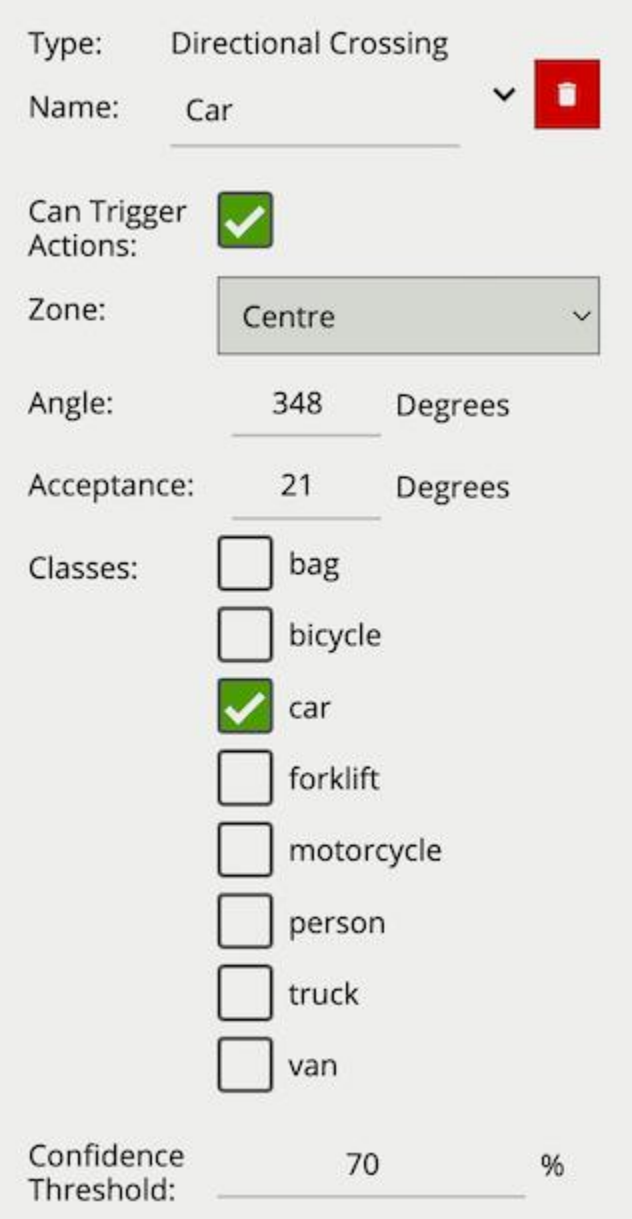
```
Res: 704x576
Input FPS: 25
Analytics: 22ms
Buffer: 120ms
Decoder: CPU
Codec: H.264
```

### 6.9.8.10.1 Graphical View



Type: Directional Crossing
Name: Car
Zone: Centre
Angle: 348
Acceptance: 21
Filters: car
Confidence Threshold: 70%
Can Trigger Actions: True

### 6.9.8.10.2 Form View



Type: Directional Crossing

Name: Car

Can Trigger Actions: ☑

Zone: Centre

Angle: 348 Degrees

Acceptance: 21 Degrees

Classes:
- □ bag
- □ bicycle
- ☑ car
- □ forklift
- □ motorcycle
- □ person
- □ truck
- □ van

Confidence Threshold: 70 %

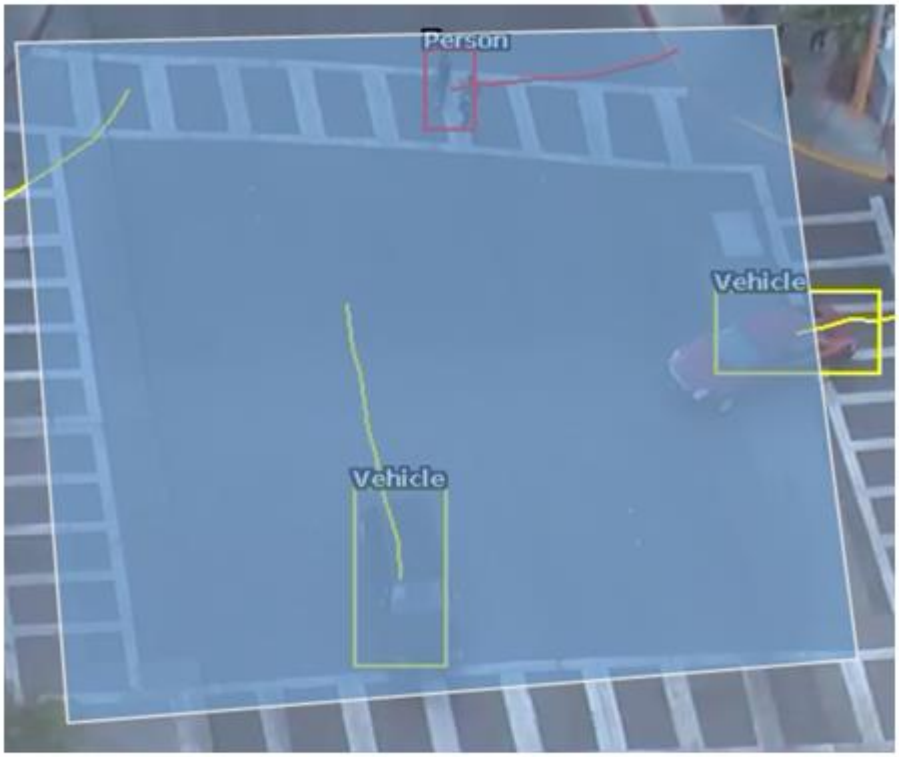
### 6.9.8.10.3 Configuration

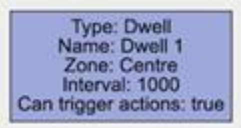
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Directional #" |

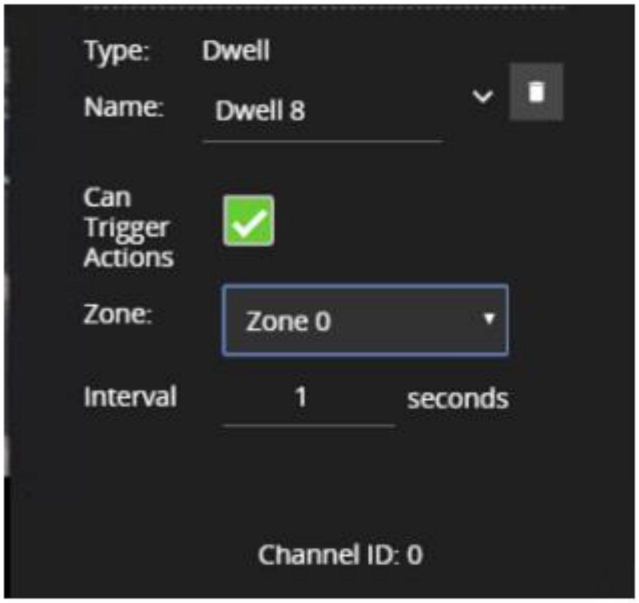
| Property | Description | Default Value |
|---|---|---|
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |
| Angle | Primary direction angle, 0 - 359. 0 references up. | 0 |
| Acceptance | Allowed variance each side of primary direction that will still trigger rule | 0 |
| Classes | The object classes allowed to trigger an alert | None |

#### *6.9.8.11 VCA - Dwell*

A dwell rule triggers when an object has remained in a zone for a specified amount of time.
The interval parameter is the time the object has to remain in the zone before an event is triggered.
The following image illustrates how the person detected in the zone is highlighted red as they have dwelt in the zone for the desired period of time.
The two vehicles have not been present in the zone for long enough yet to trigger the dwell rule.




Type: Dwell
Name: Dwell 1
Zone: Centre
Interval: 1000
Can trigger actions: true


Type: Dwell

Name: Dwell 8

Can Trigger Actions ☑

Zone: Zone 0

Interval 1 seconds

Channel ID: 0

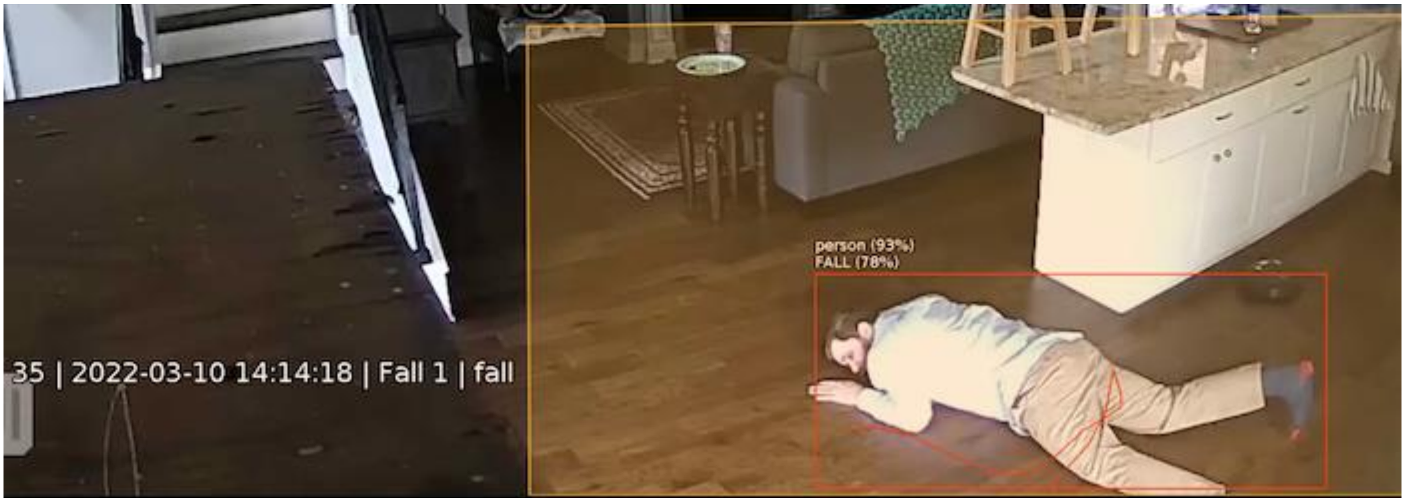
### 6.9.8.11.1 Configuration Dwell

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Direction #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |
| Interval | Period of time in seconds) | 1 |

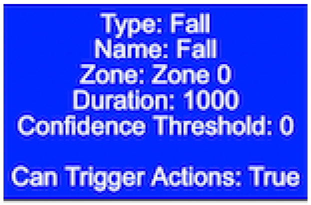
### ***6.9.8.12 VCA - Fall***

The fall rule detects when a object classified as a Person, by the Deep Learning People Tracker, is in the fallen state.

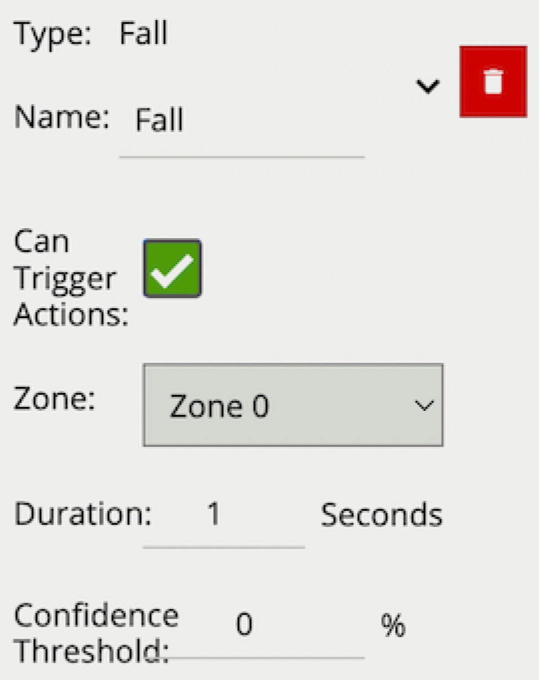
When the Fall rule is added to a channel configuration, the fall detection algorithm begins to run in the background which will have a GPU overhead. Currently this rule is only available when using the Deep Learning People Tracker.



### 6.9.8.12.1 Graphical View


Type: Fall
Name: Fall
Zone: Zone 0
Duration: 1000
Confidence Threshold: 0
Can Trigger Actions: True

### 6.9.8.12.2 Form View



Type: Fall

Name: Fall

Can Trigger Actions: ☑

Zone: Zone 0

Duration: 1 Seconds

Confidence Threshold: 0 %

### 6.9.8.12.3 Configuration

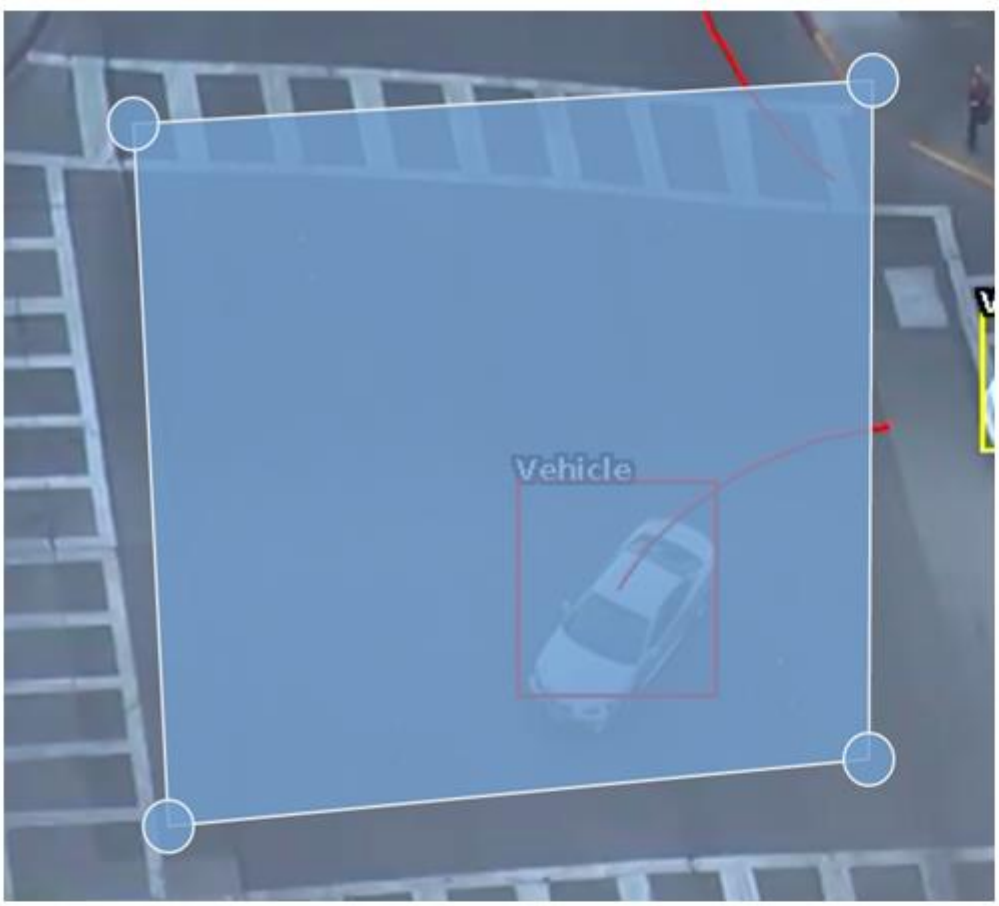
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Fall #" |
| Zone | The zone this rule is associated with | None |
| Duration | Period of time a object must have been fallen before the rule triggers | 1 to 60 seconds |
| Confidence Threshold | The algorithm confidence (as a percentage) required to trigger the rule | 0 |
| Can Trigger Actions | The algorithm confidence (as a percentage) required to trigger the rule | Active |

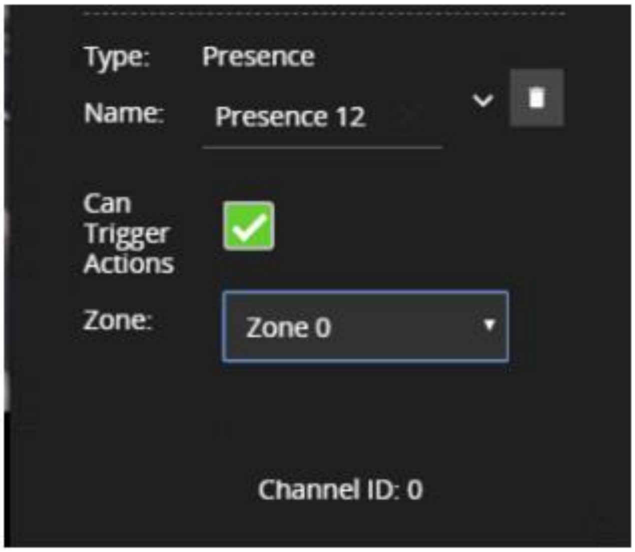
### *6.9.8.13 VCA - Presence*

A rule which fires an event when an object is first detected in a particular zone.

Note: The Presence rule encapsulates a variety of different behaviour, for example, the Presence rule will trigger in the same circumstances as an Enter and Appear rule.

The choice of which rule is most appropriate will be dependent on the scenario.




Type: Presence

Name: Presence 12

Can Trigger Actions ☑

Zone: Zone 0

Channel ID: 0

### 6.9.8.13.1 Configuration Presence

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Deep Learning Presence #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Zone | The zone this rule is associated with | None |

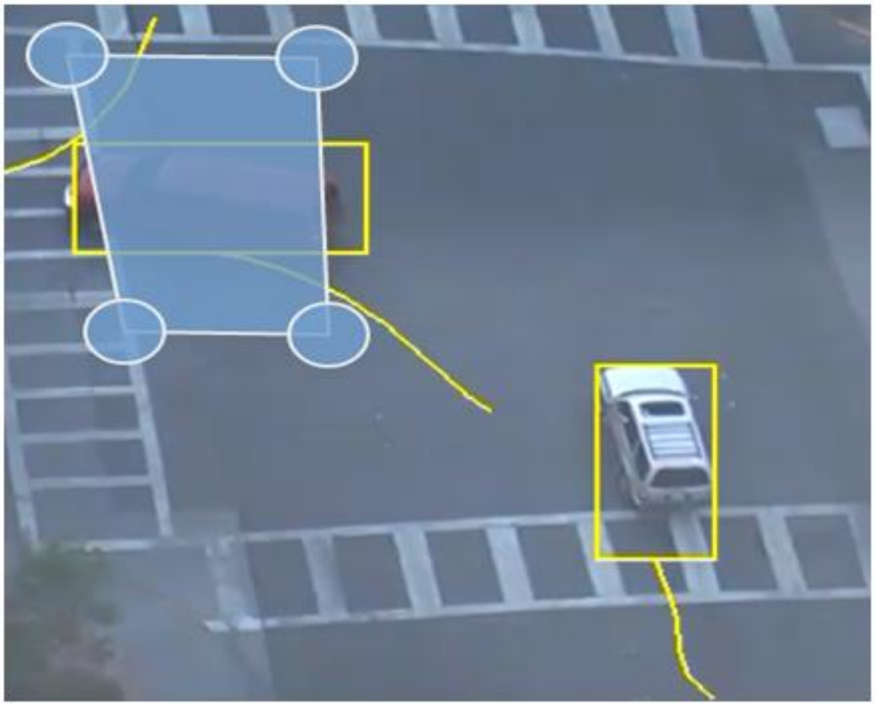
### ***6.9.8.14 VCA - Tailgating***

The tailgating rule detects objects which cross through a zone or over a line within quick succession of each other.

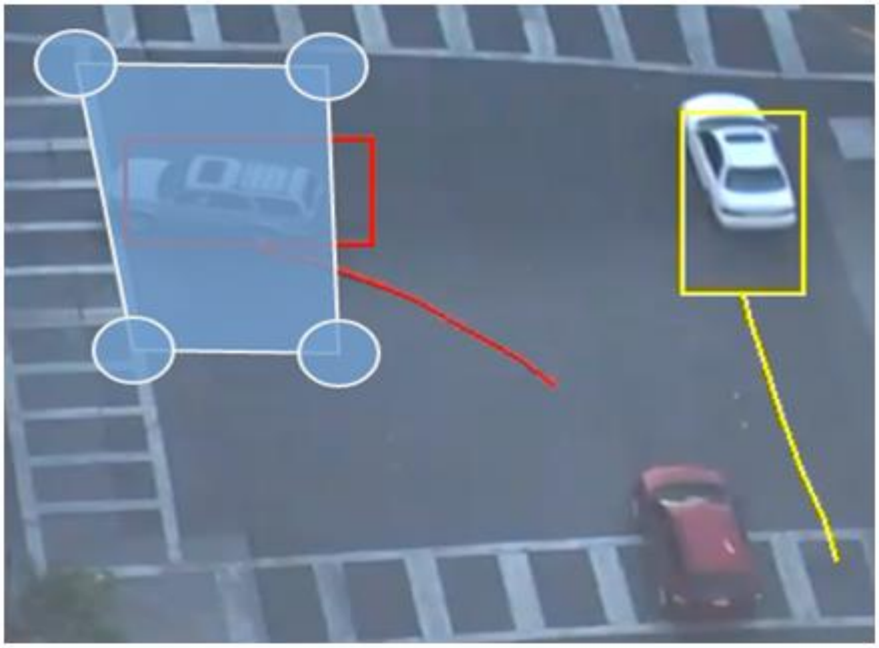
In this example, object 1 is about to cross a detection line. Another object (object 2) is following closely behind.

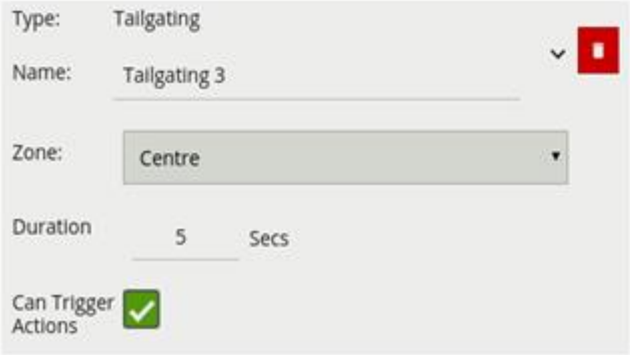
The tailgating detection threshold is set to 5 seconds.

That is, any object crossing the line within 5 seconds of an object having already crossed the line will trigger the object tailgating rule.



Object 2 crosses the line within 5 seconds of object 1. This triggers the tailgating filter and raises an event.





Type: Tailgating

Name: Tailgating 3

Zone: Centre

Duration 5 Secs

Can Trigger Actions ☑

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Tailgating #" |
| Zone | The zone this rule is associated with | None |
| Duration | The maximum amount of time between first and second object entering a zone to trigger the rule | 0 |

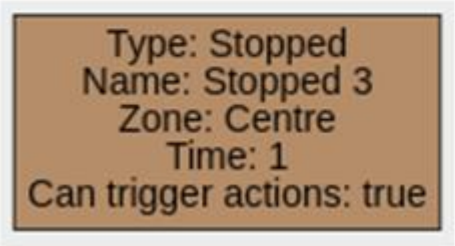
| | | |
|---|---|---|
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

### *6.9.8.15 VCA - Stopped*

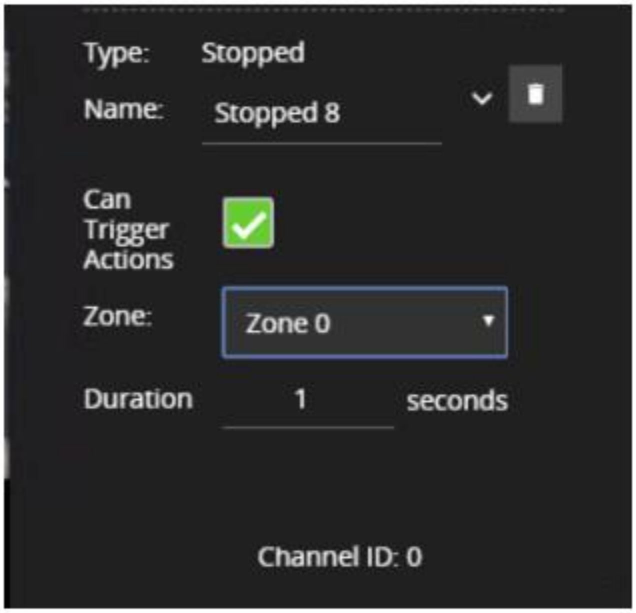
The stopped rule detects objects which are stationary inside a zone for longer than the specified amount of time.

The stopped rule requires a zone to be selected before being able to configure an amount of time.

**Note**: The stopped rule does not detect abandoned objects. It only detects objects which have moved at some point and then become stationary.



Type: Stopped
Name: Stopped 3
Zone: Centre
Time: 1
Can trigger actions: true



Type: Stopped

Name: Stopped 8

Can Trigger Actions ☑

Zone: Zone 0

Duration 1 seconds

Channel ID: 0

#### 6.9.8.15.1 Configuration Stopped

| Property | Description | Default Value |
|---|---|---|

| | | |
|---|---|---|
| Name | A user-specified name for this rule | "Stopped #" |
| Zone | The zone this rule is associated with | None |
| Time | Period of time before a stopped object triggers the rule | 0 |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

### *6.9.8.16 VCA - Counting Line*

A counting line is a detection filter optimized for directional object counting (e.g. people or vehicles) in busier detection scenarios.

**Examples of such applications may include**:

- People counting with overhead cameras in a retail environment.
- Vehicle counting with overhead cameras on public highways.

In some scenes, such as entrances with cameras installed overhead, the counting line typically will generate a higher accuracy count than using the aforementioned counters connected to a presence rule.

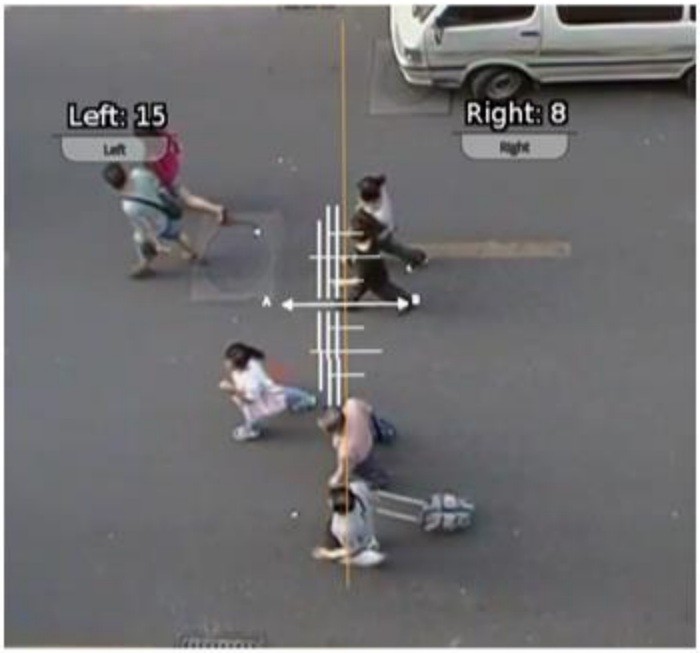
An event is generated every time an object crosses the line in the configured direction.
If multiple objects cross the line together, multiple corresponding events are generated.
These events can be directly used to trigger actions if the Can Trigger Actions property is checked.
Counting lines are attached to zones configured with a Line shape.
See **Zones** for more information.

If a counting line is configured with a zone not defined with a Line shape, the zone property will be automatically changed (it will not be possible to change the zone shape back until the counting line stops referencing the zone in question).

Counting lines have a specified direction indicated by the arrow in the UI (direction A or B), the direction of this arrow is governed by the configured zone.
Each instance of the rule counts in a single direction. To count in both directions a second counting line rule must be added to the same zone with the opposite direction selected.

An example rule graph of a two-way counting line configured with a counter is provided to illustrate this below.

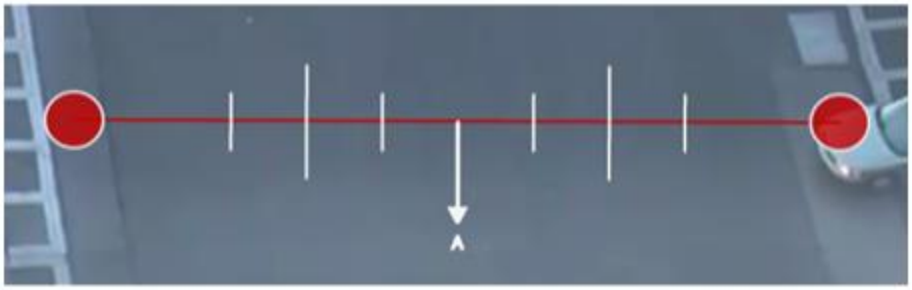
**NOTE:** The maximum number of counting line filters that can be applied per video channel is 5.



### 6.9.8.16.1 Calibrating the Counting Line

In order to generate accurate counts, the counting line requires calibration.
Unlike the object tracking function engine, this cannot be performed at a general level for the whole scene using the 3D Calibration tool.
This is because the counting line is not always placed on the ground plane; it may be placed at any orientation at any location in the scene.

For example, a counting line could be configured vertically with a side-on camera view.
Instead of the 3D calibration tool, the counting line has its own calibration setting.
Two bars equidistant from the center of the line represent the width of the expected object. This allows the counting line to reject noise and also count multiple objects.



*6.9.8.16.1.1 To calibrate the counting line:*

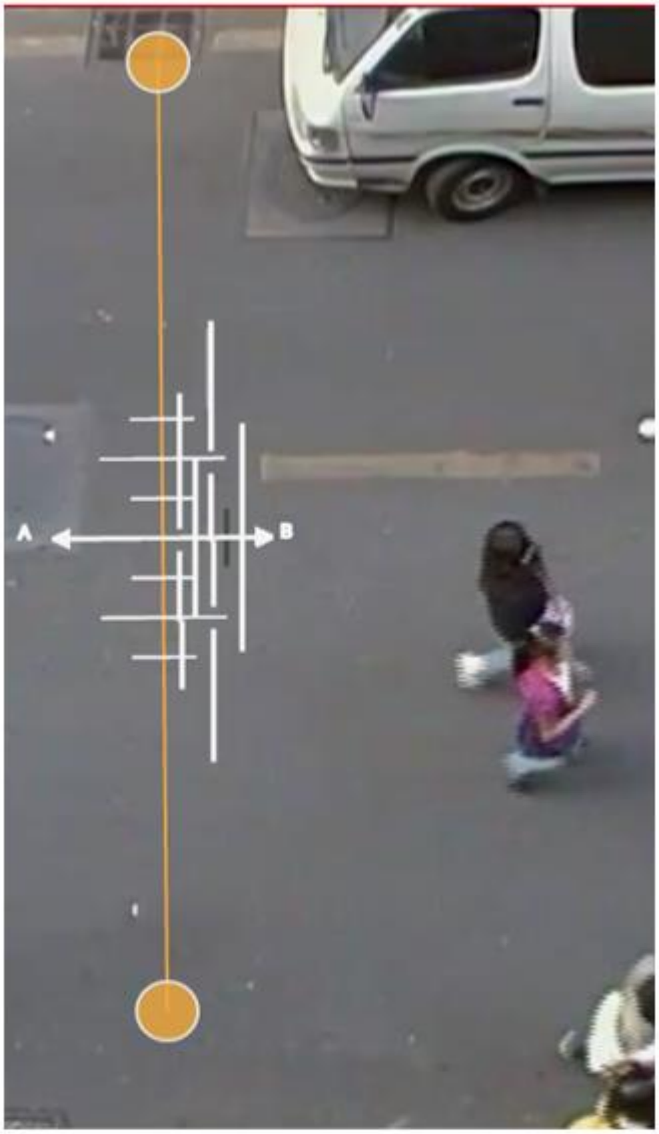
1. Select the counting line rule.
2. Check the Enable width calibration option.
3. Drag the calibration markers to adjust the distance between the calibration markers until the distance is approximately the size of the objects to be counted. Alternatively, move the Width slider to achieve the same result.
4. The calibration width is displayed within the counting line rule and can be edited directly to change the calibration width.
5. The small markers on either side of the big markers indicate the minimum and maximum width which is counted as a single object.

**NOTE:** if the Width slider is set to zero then the Enable width calibration checkbox is automatically disabled.

*6.9.8.16.1.2 Counting Line Calibration Feedback*

To enable the user to more accurately configure the calibration for the counting line, the widths of detected objects are displayed as an overlay next to the counting line when objects pass over it. By default, this display option is enabled. However, if it does not appear, ensure that the option is enabled on the Burnt-in Annotation settings.

The calibration feedback is rendered as black and white lines on either side of the counting line on the Zones configurations page.Each line represents an object detected by the counting algorithm.The width of the line shows the width of the object detected by the line.The last few detections are displayed for each direction with the latest one appearing closest to the counting line



### 6.9.8.16.2 Shadow Filter

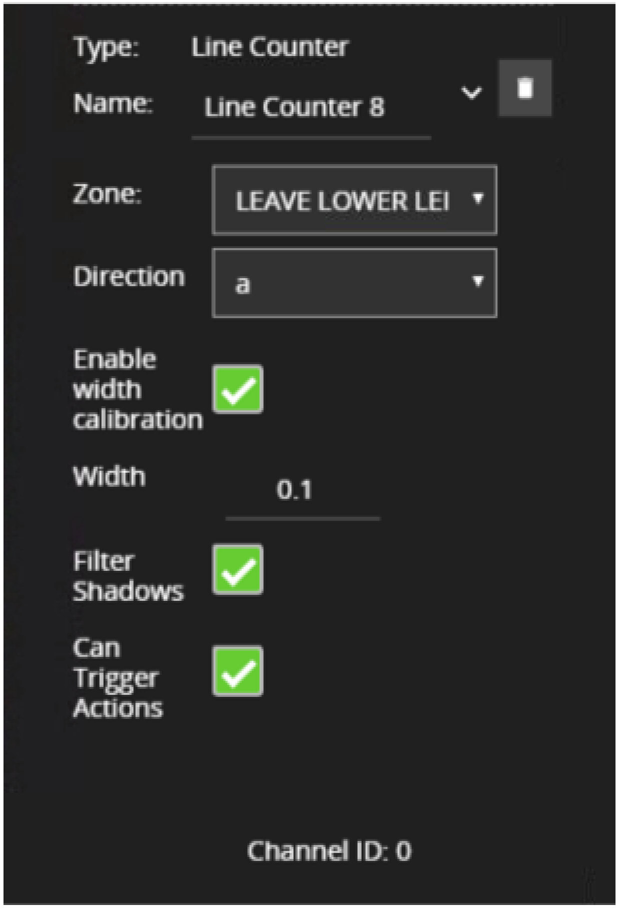
The counting line features a shadow filter which is designed to remove the effects of object shadows affecting the counting algorithm.

Shadows can cause inaccurate counting results by making an object appear larger than its true size or by joining two or more objects together.

If shadows are causing inaccurate counting, the shadow filter should be enabled by selecting the Shadow Filter check box for the line.

It is recommended that the shadow filter only be enabled when shadows are present because the

algorithm can mistake certain parts of an object for shadows and this may lead to worse counting results.
This is especially the case for objects that have little contrast compared to the background (e.g. people wearing black coats against a black carpet).



Type: Line Counter

Name: Line Counter 8

Zone: LEAVE LOWER LEI ▾

Direction a ▾

Enable width calibration ☑

Width 0.1

Filter Shadows ☑

Can Trigger Actions ☑

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | Line_Counter |
| Zone | The zone this rule is associated with | None |
| Direction | Enable counting in the 'A' or 'B' direction (one direction per counting line) | None |

| | | |
|---|---|---|
| Enable Width Calibration | Width calibration to allow more accurate counting | None |
| Width | Width calibration value | 0 |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

### ***6.9.8.17 VCA - Conditional rules***

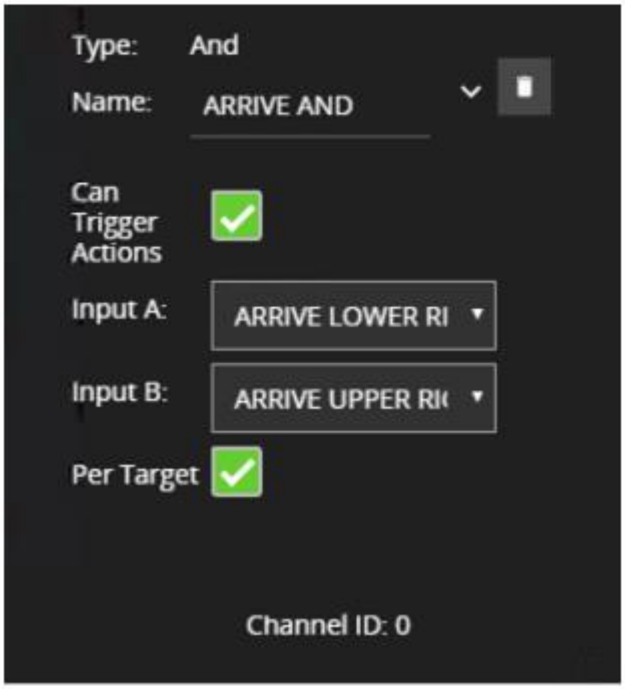
A conditional input, like a filter, is one that cannot trigger an action on its own.
It requires the input of another basic input, conditional rule or filter to be meaningful.
An example of this is the AND rule. The AND rule requires two inputs to compare in order to function.

#### 6.9.8.17.1 The complete list of conditional rules are:

- And
- Continuously
- Counter
- Or
- Previous

#### 6.9.8.17.2 And

A logical operator that combines two rules and only fires events if both inputs are true.



Type: And

Name: ARRIVE AND

Can Trigger Actions ☑

Input A: ARRIVE LOWER RI

Input B: ARRIVE UPPER RI(

Per Target ☑

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "And #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input A | The first input | None |
| Input B | The second input | None |
| Per Target | Fire one event per tracked object | Active |

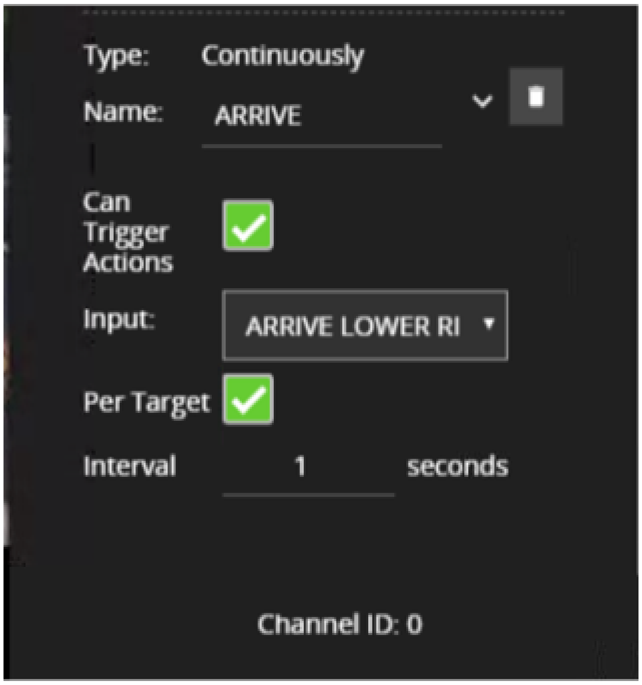
If we consider a scene with two presence rules, connected to two separate zones, connected by an AND rule, the table below explains the behaviour of the Per Target property.
Note that object here refers to a tracked object, as detected by the VCA tracking engine.

| State | Per Target | Outcome |
|---|---|---|
| Object A in Input A, Object B in input B | On | Two events were generated, one for each object |
| Object A in Input A, Object B in input B | Off | Only one event generated |

Additionally, it is important to note that if the rule fires when Per Target is switched off, it will not fire again until it is 'reset', i.e. until the AND condition is no longer true.

### 6.9.8.17.3 Continuously

A logical operator fires events when its input has occurred continuously for a user-specified time.



Type: Continuously

Name: ARRIVE

Can Trigger Actions ☑

Input: ARRIVE LOWER RI ▾

Per Target ☑

Interval 1 seconds

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Continuously #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |
| Per Target | Fire one event per tracked object. See the description below for more details | Active |

| | | |
|---|---|---|
| Interval | The time in milliseconds | 1000 ms |

Considering a scene with one zone, a presence rule associated with that zone, and a Continuously rule attached to that presence rule, when the Per Target property is on, the rule will generate an event for each tracked object that is continuously present in the zone.

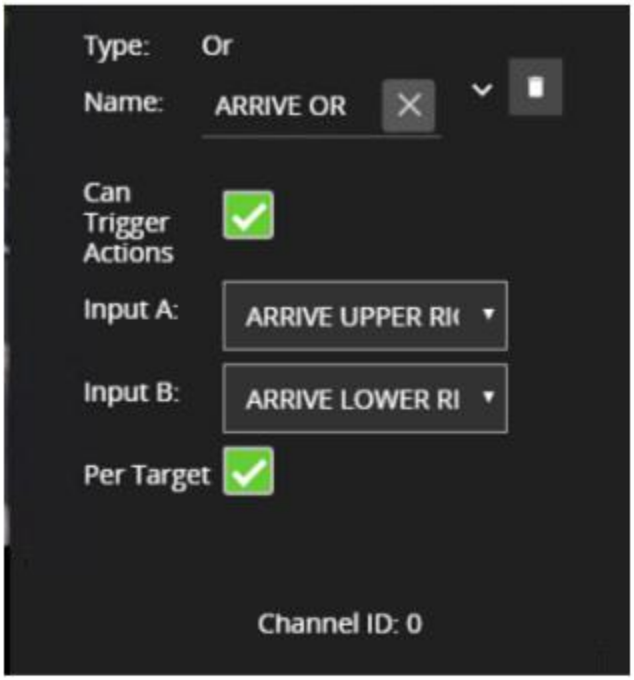
When it is off, only one event will be generated by the rule, even if there are multiple tracked objects within the zone.
Additionally, when Per Target is off, the rule will only generate events when there is a change of state - i.e. the rule condition changes from true to false or vice versa.

#### *6.9.8.17.3.1 When Per Target is off, the state will change when:*

1. Any number of objects enter the zone in question and remain in the zone
2. All objects leave the zone in question

### 6.9.8.17.4 Or

A logical operator that combines two rules and fires events if either input is true.



Type: Or
Name: ARRIVE OR
Can Trigger Actions ☑
Input A: ARRIVE UPPER RI(
Input B: ARRIVE LOWER RI
Per Target ☑
Channel ID: 0

| Property | Description | Default Value |
|---|---|---|

| | | |
|---|---|---|
| Name | A user-specified name for this rule | "Or #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input A | The first input | None |
| Input B | The second input | None |
| Per Target | Fire one event per tracked object | Active |

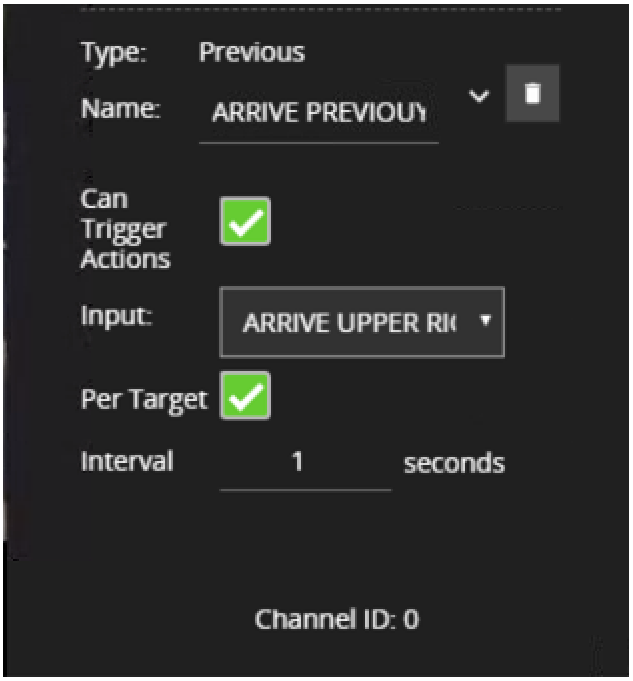
If we consider a scene with two presence rules, connected to two separate zones, connected by an OR rule, the table below explains the behaviour of the Per Target property.

| State | Per Target | Outcome |
|---|---|---|
| Object A in Input A, No object in input B | On | Two events were generated, one for each object |
| No object in Input A, Object B in input B | On | Only one event was generated (for Object B) |
| Object A in Input A, No object in input B | On | Only one event generated (for Object A) |
| Object A in Input A, No object in input B | Off | Only one event generated |
| No object in Input A, Object B in input B | Off | Only one event generated |
| Object A in Input A, No object in input B | Off | Only one event generated |

### 6.9.8.17.5 Previous

A logical operator triggers for input events that were active at some point in a past window of time.

This window is defined by between the current time and the period before the current time (specified by the interval parameter value).



Type: Previous

Name: ARRIVE PREVIOUY

Can Trigger Actions ☑

Input: ARRIVE UPPER RI( ▾

Per Target ☑

Interval 1 seconds

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Previous #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |
| Per Target | Fire one event per tracked object | Active |
| Interval | The time in milliseconds | 1000 ms |

### *6.9.8.18 VCA - Counters*

These counters are only visible in the VCA configuration. To use counters in the Spotter please refer to the Spotter manual.

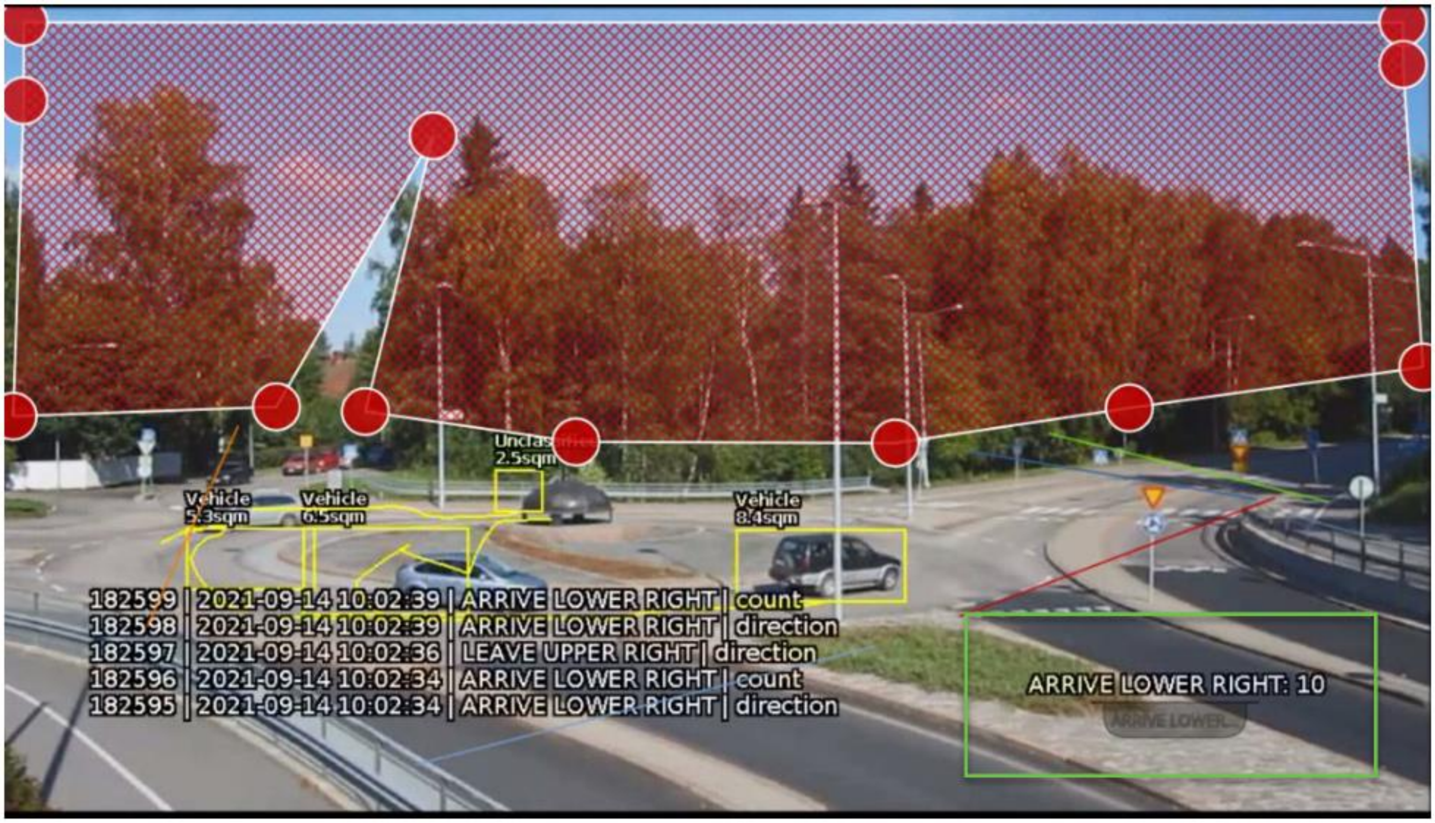
Counters can be configured to count the number of times a rule is triggered, for example, the number of people crossing a line.

The counter rule is designed to be utilized in two ways:

- Increment / Decrement: whereby a counter is incremented by the attached rule(s) (+1 for each rule trigger) and decremented by another attached rule(s) (-1 for each rule trigger).
- Occupancy: whereby the counter reflects the number of objects that are currently triggering the attached rule(s).

More than one rule can be assigned to any of a counter's three inputs.
This allows, for example, the occupancy of two presence rules to be reflected in a single counter or more than one entrance/exit gate to reflect in a single counter, an example rule graph is provided to illustrate this below.
Broadly speaking a single counter should not be used for both purposes occupancy and increment/decrement.

Note: events created by a counter will not trigger the Deep-Learning Filter, even if enabled on the channel.

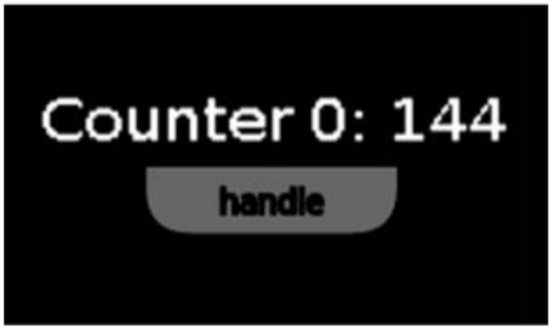


```
182599 | 2021-09-14 10:02:39 | ARRIVE LOWER RIGHT | count
182598 | 2021-09-14 10:02:39 | ARRIVE LOWER RIGHT | direction
182597 | 2021-09-14 10:02:36 | LEAVE UPPER RIGHT | direction
182596 | 2021-09-14 10:02:34 | ARRIVE LOWER RIGHT | count
182595 | 2021-09-14 10:02:34 | ARRIVE LOWER RIGHT | direction
```

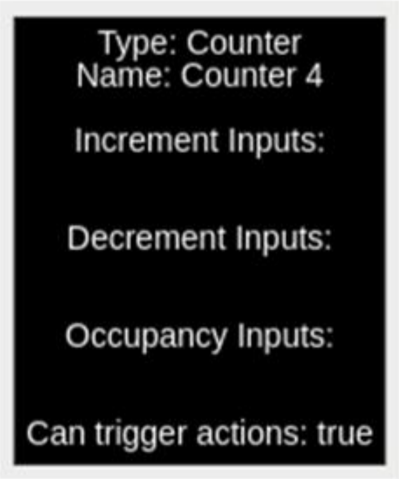
#### 6.9.8.18.1 Positioning Counters

When added, a counter object is visualized on the video stream as seen below.

The counter can be repositioned by grabbing the 'handle' beneath the counter name and moving the counter to the desired location.



Counter 0: 144



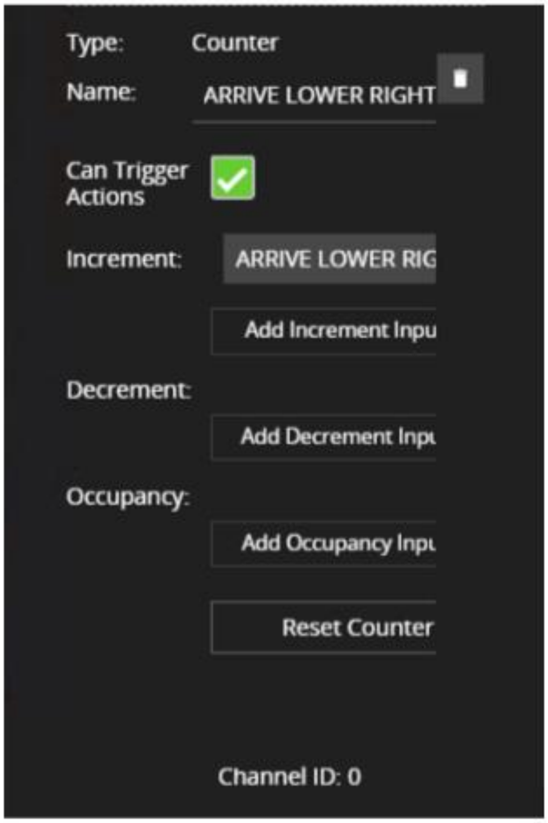
Type: Counter
Name: Counter 4

Increment Inputs:

Decrement Inputs:

Occupancy Inputs:

Can trigger actions: true



Type: Counter

Name: ARRIVE LOWER RIGHT

Can Trigger Actions ☑

Increment: ARRIVE LOWER RIG

Decrement:

Occupancy:

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Counter #" |
| Increment | The rule which, when triggered, will add one to the counter | None |
| Decrement | The rule which, when triggered, will subtract one from the counter | None |
| Occupancy | Sets counter to the current number of the rule's active triggers* | None |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Reset Counter | A button allowing the counter value to be reset to 0 | None |

- E.g. if a presence rule is set as the occupancy target and two objects are currently triggering that presence rule, the counter will show the value of '2'.

***6.9.8.19 VCA - Object trails***

The trail shows the history of where the object has been.

Depending on the calibration the trail can be drawn from the centroid or the mid-bottom point of the object.

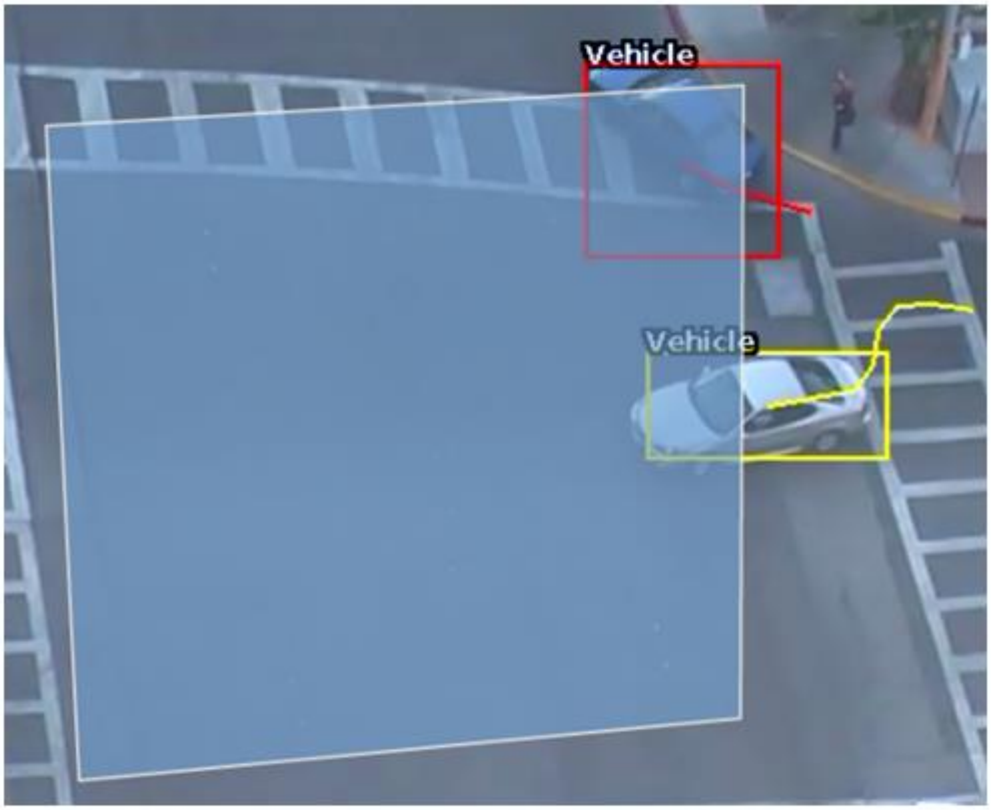
(See Advanced Settings for more information).

The trail is important for determining how a rule is triggered.

The intersection of the trail point with a zone or line determines whether a rule is triggered or not.

The following image illustrates this point: the blue vehicle's trail intersects with the detection zone and is rendered in red.

Conversely, while the white vehicle intersects the detection zone, its trail does not (yet) intersect and hence it has not triggered the rule and is rendered in yellow.



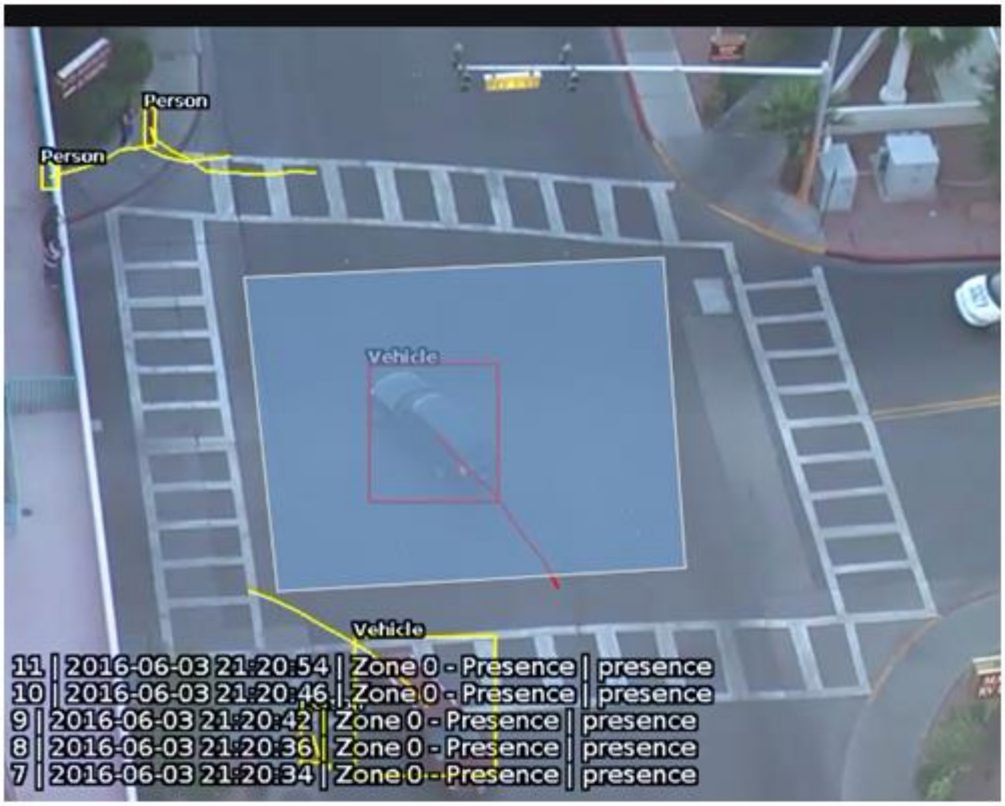
### 6.9.8.20 VCA - Object display

As rules are configured, they are applied to the channel in real-time allowing feedback on how they work.
Objects which have triggered a rule are annotated with a bounding box and a trail. Objects can be rendered in two states:

1. **Non-alarmed:** Default rendered in yellow. A detected object which does not meet any criteria trigger a rule and raise an event.

2. **Alarmed:** Default rendered in red. A detected object which has triggered one or more rules. Causes an event to be raised.

As seen below, when an event is raised, the default settings render details of the event in the lower half of the video stream.
Object class annotations in this example are generated through



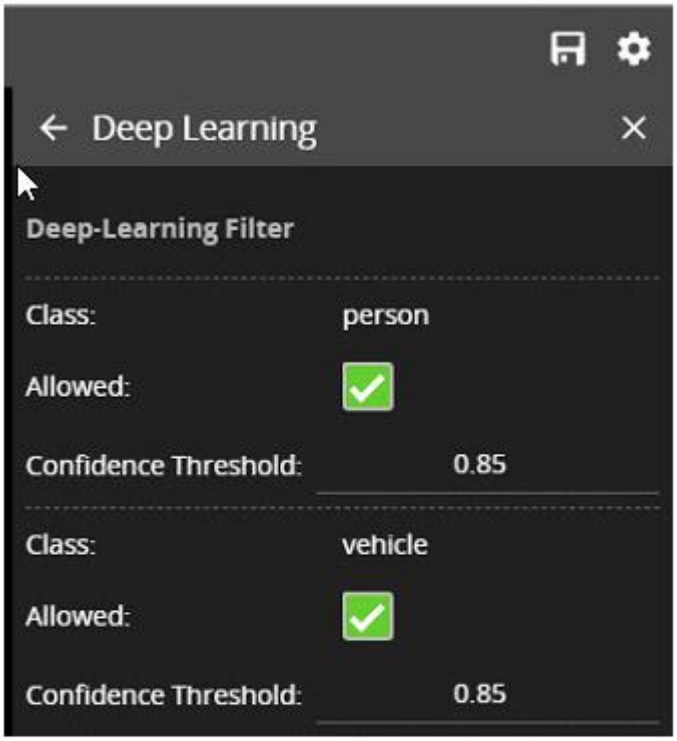
```
11 | 2016-06-03 21:20:54 | Zone 0 - Presence | presence
10 | 2016-06-03 21:20:46 | Zone 0 - Presence | presence
9 | 2016-06-03 21:20:42 | Zone 0 - Presence | presence
8 | 2016-06-03 21:20:36 | Zone 0 - Presence | presence
7 | 2016-06-03 21:20:34 | Zone 0 - Presence | presence
```

### 6.9.9 VCA - Deep-Learning Filter

VCAserver also supports classification through the use of the deep learning filter. In this case an object, which has triggered a rule, can be analyzed using the deep learning filter and a predicted class and confidence level returned. The available object classes are defined by the model.

On VCAserver the Deep Learning Filter can use GPU acceleration, see **Deep Learning Requirements** for hardware requirements.

Without GPU acceleration the deep learning filter will use the CPU, enabling the deep learning filter on multiple channels which are generating a high volume of events, (more than 1 per second) may result in poor performance of the system and is not advised.



Deep-Learning Filter

Class: person

Allowed: ☑

Confidence Threshold: 0.85

Class: vehicle

Allowed: ☑

Confidence Threshold: 0.85

***6.9.9.1 Each of the possible object classes has additional parameters*:**

**Allowed**: Whether this object type will be allowed to pass through the filter. If this is unchecked, any objects classified as this type will not trigger any actions.

**Confidence Threshold**: A value (0.0 - 1.0) representing the minimum confidence level required for classification. Any objects with a lower classification score than this minimum value will be filtered out and will not trigger any actions.

## 6.10 VCA – Filters

A filter cannot trigger an action on its own as it requires another basic input, filter or conditional rule to trigger.

- VCA Filters - Speed Filter
- VCA Filters - Object Filter

- VCA Filters - Colour Filter
- VCA Filters - Retrigger Filter
- VCA Filters - Deep Learning Filter

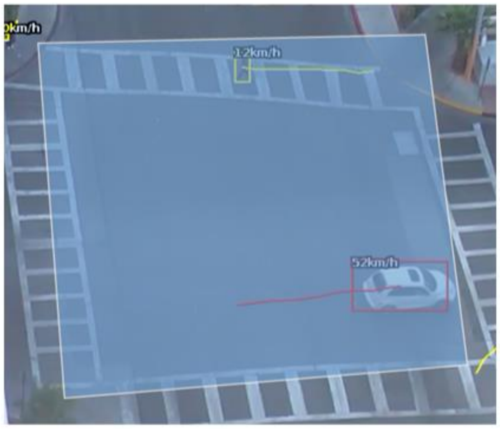
### 6.10.1 Speed Filter

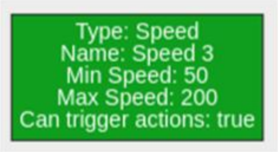
The speed filter provides a way to check if the speed of an object which has triggered an input is moving within the range of speeds defined by a lower and upper boundary.

> The channel must be Calibrated in order for the speed filter to be available.

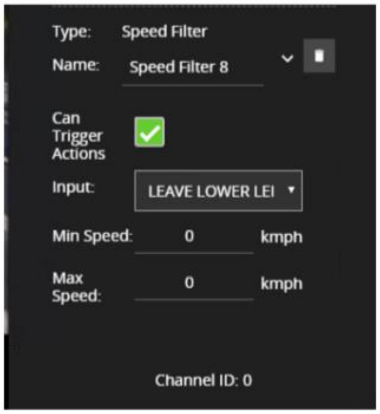
Commonly this rule is combined with a presence rule, an example rule graph is provided to illustrate this below.

The following image illustrates how such a rule combination triggers on the car moving at 52 km/h but the person moving at 12 km/h falls outside the configured range (25-100 km/h) and thus does not trigger the rule.





Type: Speed
Name: Speed 3
Min Speed: 50
Max Speed: 200
Can trigger actions: true



Type: Speed Filter

Name: Speed Filter 8

Can Trigger Actions ☑

Input: LEAVE LOWER LEI ▾

Min Speed: 0 kmph

Max Speed: 0 kmph

Channel ID: 0

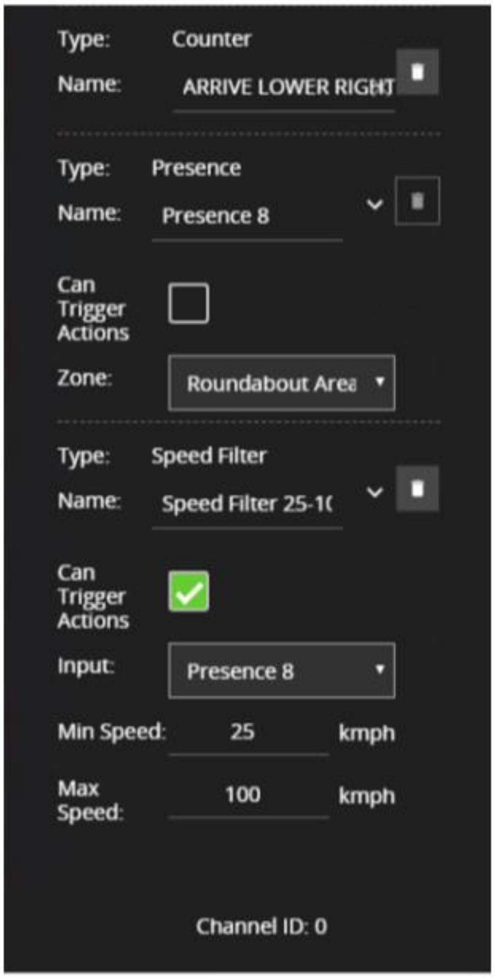
| **Property** | **Description** | **Default Value** |
|---|---|---|
| Name | A user-specified name for this rule | "Speed #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |

| Property | Description | Default Value |
|---|---|---|
| Min Speed | The minimum speed (km/h) an object must be going to trigger the rule | 0 |
| Max Speed | The maximum speed (km/h) an object can be going to trigger the rule | 0 |

### *6.10.1.1 Typical Logical Rule Combination*

The below example logical rule checks if an object triggering the presence rule attached to zone Centre, is also travelling between 25 and 100 km/h as specified by the speed rule Speed Filter 25-100 km/h.

Only the Speed Filter is set to Can Trigger Actions, meaning only this component of the logical rule will be available as a source for actions. Additionally, any activity generated by the speed filter will have the event type Presence



Type: Counter
Name: ARRIVE LOWER RIGHT

Type: Presence
Name: Presence 8
Can Trigger Actions ☐
Zone: Roundabout Area

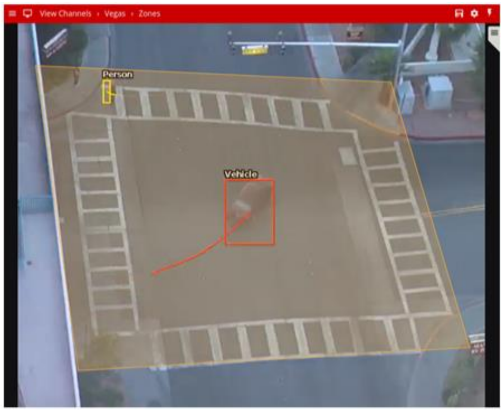
Type: Speed Filter
Name: Speed Filter 25-1(
Can Trigger Actions ☑
Input: Presence 8
Min Speed: 25 kmph
Max Speed: 100 kmph

Channel ID: 0

### 6.10.2 VCA Filters - Object Filter

The object classification filter provides the ability to filter out objects, which trigger a rule if they are not classified as a certain class (e.g. person, vehicle).

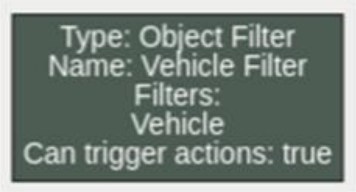
The object classification filter must be combined with another rule(s) to prevent unwanted objects from triggering an alert, an example rule graph is provided to illustrate this below.



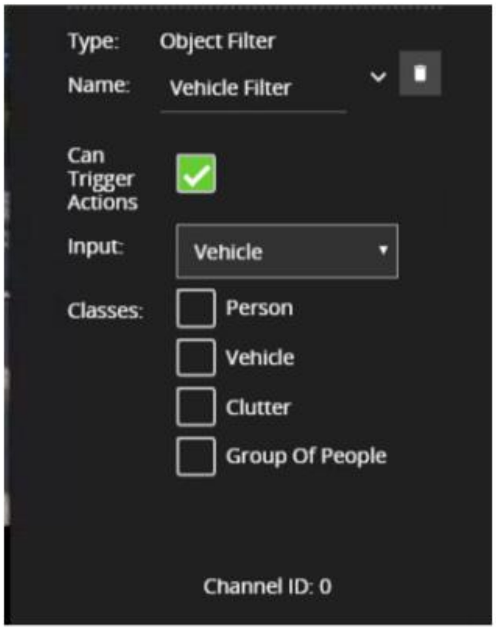
The previous image illustrates how the object classification filter configured with Vehicle class, includes only Vehicle objects.

The person in the zone is filtered out since the Person class is not selected in the filter list.

> The channel must be calibrated for the object classification filter to be available.



Type: Object Filter
Name: Vehicle Filter
Filters:
Vehicle
Can trigger actions: true



Type: Object Filter

Name: Vehicle Filter

Can Trigger Actions ☑

Input: Vehicle

Classes: □ Person
□ Vehicle
□ Clutter
□ Group Of People

Channel ID: 0

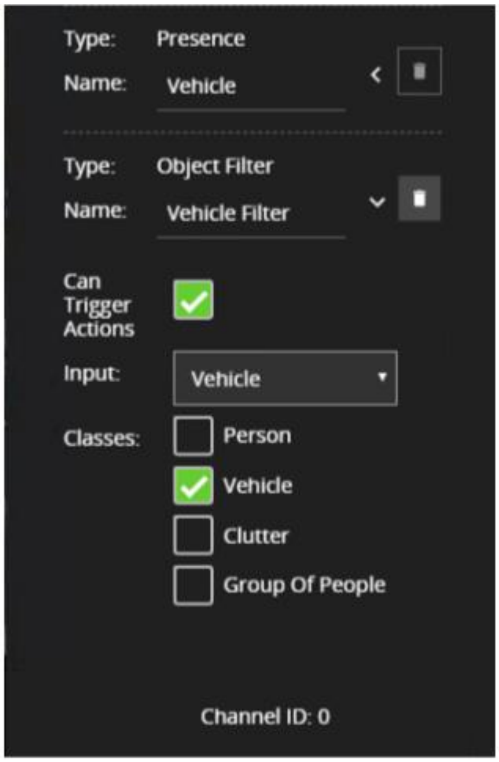
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Object Filter #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |
| Classes | The object classes allowed to trigger an alert | None |

#### *6.10.2.1 Typical Logical Rule Combination*

The below example logical rule checks if the object triggering the presence rule attached to zone Centre, is also classified as a Vehicle as specified by the Object Filter Vehicle Filter.

Only the Object filter is set to Can Trigger Actions, meaning only this component of the logical rule will be available as a source for actions.

Additionally, any activity generated by the speed filter will have the event type Presence.



Type: Presence
Name: Vehicle

Type: Object Filter
Name: Vehicle Filter

Can Trigger Actions ☑

Input: Vehicle

Classes:
- □ Person
- ☑ Vehicle
- □ Clutter
- □ Group Of People

Channel ID: 0

### 6.10.3 VCA Filters - Colour Filter

The colour filter utilizes the Colour Signature algorithm and provides the ability to filter out objects based on whether that object contains a certain colour component.

The colour signature algorithm is responsible for grouping every pixel from a detected object into one of 10 colour bins.

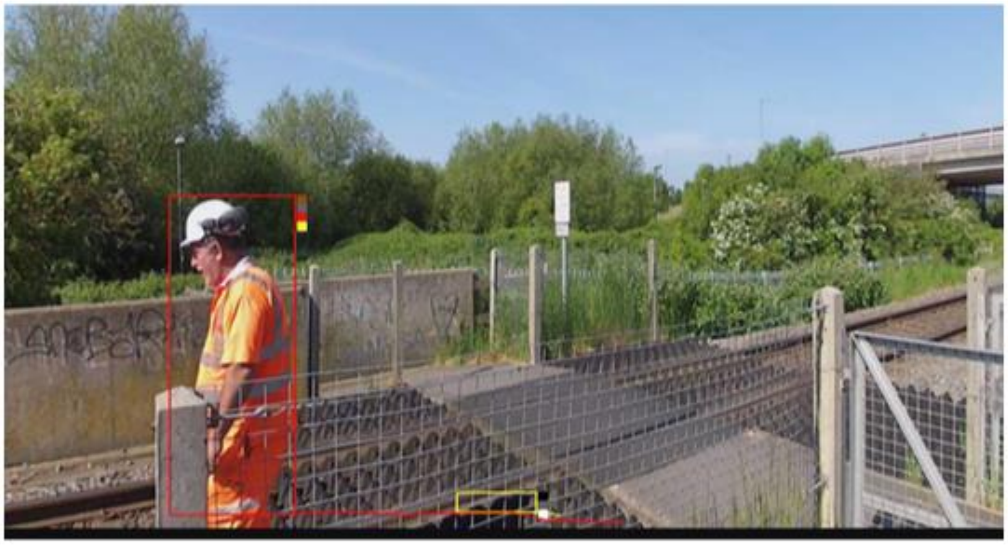
The colour filter allows you to select one or more of these colour bins and will trigger if the subject-object is made up of one or more of those selected colours.

The below image shows an example tracked object with the colour signature annotations enabled.

Here the top four colours which make up more than 5% of the object are represented by the colour swatch attached to the object.

In this case, a person is being tracked in the scene with high visibility safety clothing. Here the colour filter is set to trigger on Yellow, detecting the person but ignoring the shadow.

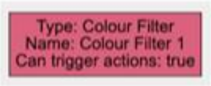
Typically, the colour classification filter would be combined with another rule(s) to prevent unwanted objects from triggering an alert, an example rule graph is provided to illustrate this below.



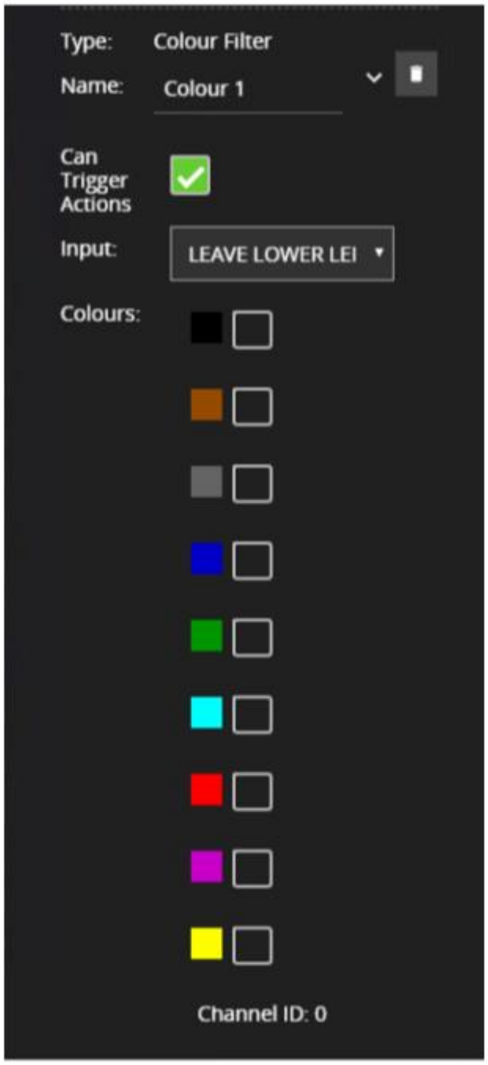
The previous image illustrates how the object classification filter configured with Vehicle class, includes only Vehicle objects.

The person in the zone is filtered out since the Person class is not selected in the filter list.

> The channel must have the Colour Signature enabled for the colour filter to work.



Type: Colour Filter
Name: Colour Filter 1
Can trigger actions: true



Type: Colour Filter

Name: Colour 1

Can Trigger Actions ☑

Input: LEAVE LOWER LEI ▾

Colours:

⬛ ☐

🟫 ☐

◼ ☐

🟦 ☐

🟩 ☐

🟦 ☐

🟥 ☐

🟪 ☐

🟨 ☐

Channel ID: 0

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Object Filter #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

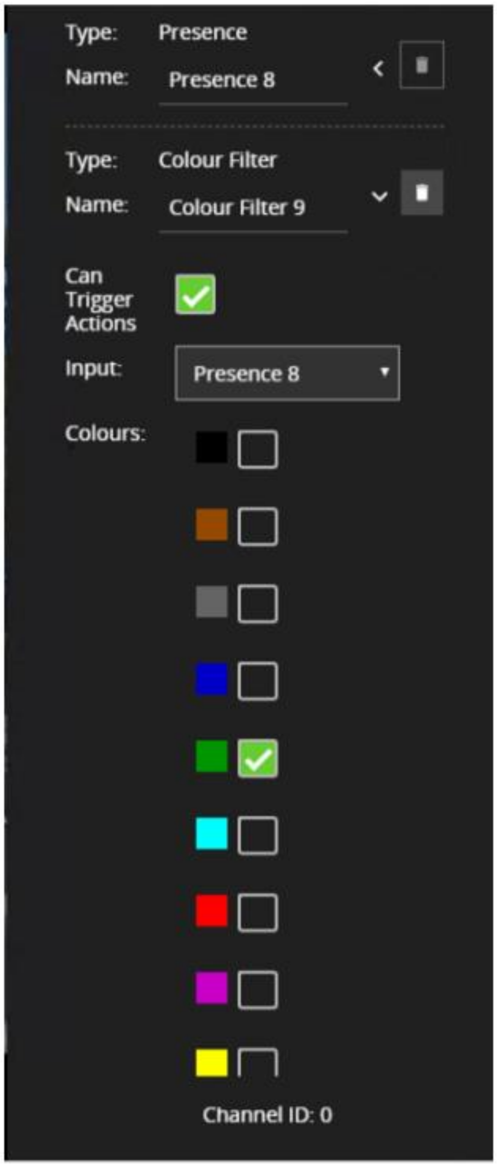
| Property | Description | Default Value |
|---|---|---|
| Input | The input rule | None |
| Colours | The colours allowed to trigger an alert | All Unchecked |

#### *6.10.3.1 Typical Logical Rule Combination*

The below example logical rule checks if the object triggering the presence rule Train line attached to zone Centre, also contains the colour Green as one of the top four colours by percentage.

Only the Colour filter is set to Can Trigger Actions, meaning only this component of the logical rule will be available as a source for actions.

Additionally, any activity generated by the speed filter will have the event type Presence.



Type: Presence
Name: Presence 8

Type: Colour Filter
Name: Colour Filter 9

Can Trigger Actions ☑

Input: Presence 8

Colours:

⬛ □
🟫 □
◼ □
🟦 □
🟩 ☑
🟦 □
🟥 □
🟪 □
🟨 □

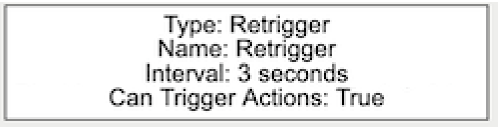
Channel ID: 0

### 6.10.4 VCA Filters - Retrigger Filter

The Retrigger Filter acts as an event pass through, which only generates an event if the input has not fired previously within the defined interval.

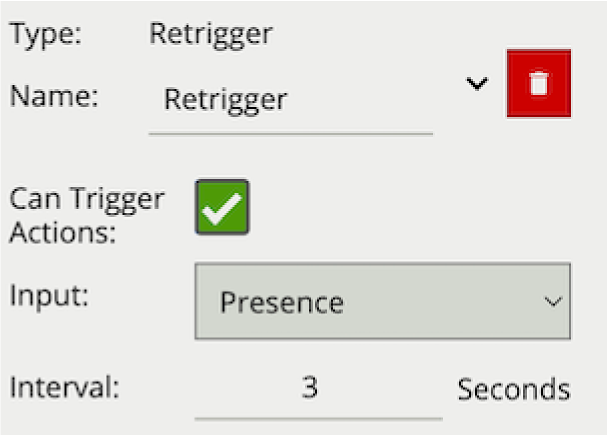
Typically, the Retrigger Filter would be applied at the end of a rule(s) combination to prevent duplicate alarms being sent, this provides more granular control than the Event Retrigger Time option. Events produced by the Retrigger Filter will have the event type of the input rule.

#### *6.10.4.1 Graphical View*



Type: Retrigger
Name: Retrigger
Interval: 3 seconds
Can Trigger Actions: True

#### *6.10.4.2 Form View*



Type: Retrigger
Name: Retrigger
Can Trigger Actions: ☑
Input: Presence
Interval: 3 Seconds

#### *6.10.4.3 Configuration*

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Retrigger #” |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |
| Interval | Period in which the input event cannot generate another event | 3 |

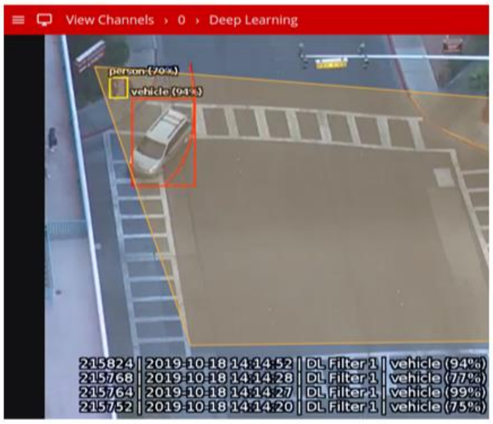
### 6.10.5 VCA Filters - Deep Learning Filter

The deep learning filter provides the ability to filter out objects, which trigger a rule if they are not classified as a certain class by the deep learning model.

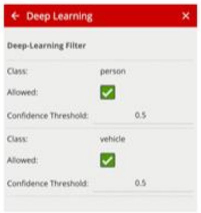
The deep learning filter settings are configured on the Deep Learning page.

See Deep Learning Filter for an in-depth description of how the filter works.

Typically the deep learning filter would be combined with another rule(s) to prevent unwanted objects from triggering an alert, an example rule graph is provided to illustrate this below. Please note that the deep learning filter cannot be used as an input to any other rule type. As such it must be the last rule in a graph



215824 | 2019-10-18 14:14:52 | DL Filter 1 | vehicle (94%)
215768 | 2019-10-18 14:14:28 | DL Filter 1 | vehicle (77%)
215764 | 2019-10-18 14:14:27 | DL Filter 1 | vehicle (99%)
215752 | 2019-10-18 14:14:20 | DL Filter 1 | vehicle (75%)

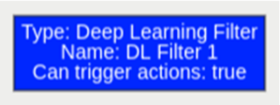


Deep-Learning Filter

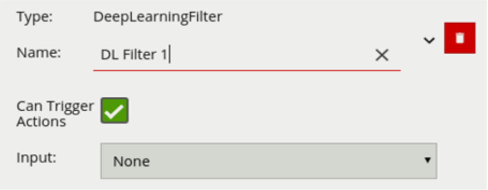
Class: person
Allowed: ☑
Confidence Threshold: 0.5
Class: vehicle
Allowed: ☑
Confidence Threshold: 0.5

The previous image illustrates how the deep learning filter configured with just vehicle class (Confidence Threshold 0.5), only triggers on the vehicle object.

The person in the zone is filtered out since the person class Allowed setting is not enabled in the Deep Learning configuration page.



Type: Deep Learning Filter
Name: DL Filter 1
Can trigger actions: true



Type: DeepLearningFilter
Name: DL Filter 1
Can Trigger Actions ☑
Input: None

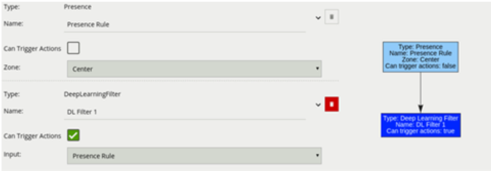
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "DL Filter #" |

| Property | Description | Default Value |
|---|---|---|
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |

### *6.10.5.1 Typical Logical Rule Combination*

The below example logical rule checks if the object triggering the presence rule attached to zone Centre, is one of the classes of interest defined in the Deep Learning settings page (see above settings page image).
Only the deep learning filter is set to Can Trigger Actions, meaning only this component of the logical rule will be available as a source for actions.

Additionally, any activity generated by the speed filter will have the event type Presence.



Type: Presence
Name: Presence Rule
Can Trigger Actions ☐
Zone: Center

Type: DeepLearningFilter
Name: DL Filter 1
Can Trigger Actions ☑
Input: Presence Rule

Type: Presence
Name: Presence Rule
Zone: Center
Can trigger actions: false

Type: Deep Learning Filter
Name: DL Filter 1
Can trigger actions: true

## 6.11 Conditional Rule Types

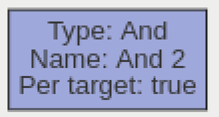
The currently supported conditional rules, along with a detailed description of each.

- And
- Continuously
- Counter
- Not
- Or
- Previous
- Repeatedly

### 6.11.1 And

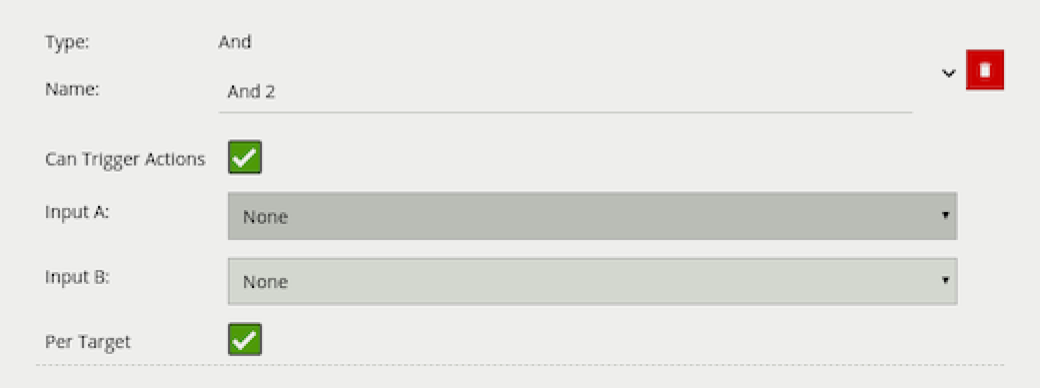
A logical operator that combines two rules and only fires events if both inputs are true.

#### *6.11.1.1 Graphical View*



Type: And
Name: And 2
Per target: true

#### *6.11.1.2 Form View*



| | |
|---|---|
| Type: | And |
| Name: | And 2 |
| Can Trigger Actions | ☑ |
| Input A: | None |
| Input B: | None |
| Per Target | ☑ |

#### *6.11.1.3 Configuration*

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "And #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input A | The first input | None |
| Input B | The second input | None |
| Per Target | Fire one event per tracked object | Active |

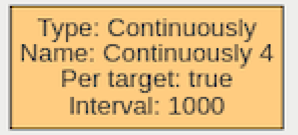
If we consider a scene with two presence rules, connected to two separate zones, connected by an AND rule, the table below explains the behaviour of the Per Target property. Note that object here refers to a tracked object, as detected by the VCA tracking engine.

| State | Per Target | Outcome |
|---|---|---|
| Object A in Input A, Object B in input B | On | Two events generated, one for each object |
| Object A in Input A, Object B in input B | Off | Only one event generated |

### 6.11.2 Continuously

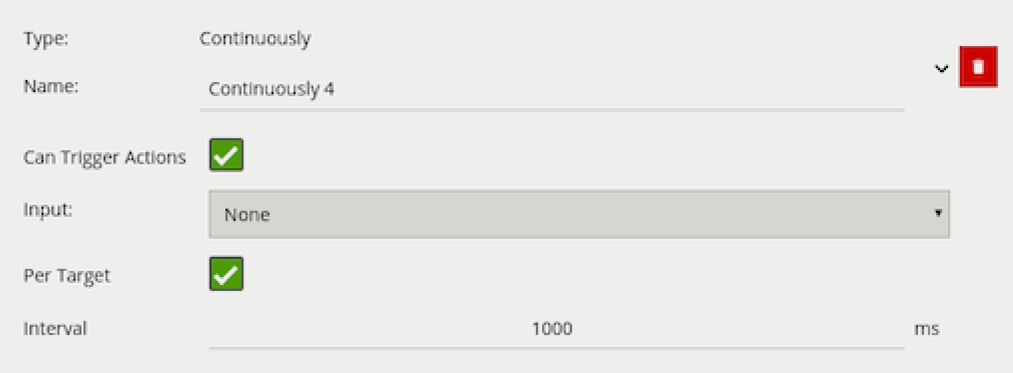
A logical operator that fires events when its input has occurred continuously for a user-specified time.

#### *6.11.2.1 Graphical View*



Type: Continuously
Name: Continuously 4
Per target: true
Interval: 1000

#### *6.11.2.2 Form View*



| | |
|---|---|
| Type: | Continuously |
| Name: | Continuously 4 |
| Can Trigger Actions | ☑ |
| Input: | None |
| Per Target | ☑ |
| Interval | 1000 ms |

#### *6.11.2.3 Configuration*

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Continuously #” |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |

| Property | Description | Default Value |
|---|---|---|
| Per Target | Fire one event per tracked object. See description below for more details | Active |
| Interval | The time in milliseconds | 1 |

Considering a scene with a Presence rule associated with a zone and a Continuously rule attached to that Presence rule, when the **Per Target** property is on, the rule will generate an event for each tracked object that is continuously present in the zone. When it is off, only one event will be generated by the rule, even if there are multiple tracked objects within the zone. Additionally, when **Per Target** is off, the rule will only generate events when there is change of state, i.e. the rule condition changes from true to false or vice versa. When **Per Target** is off, the state will change when:

- Any number of objects enter the zone in question and remain in the zone
- **All** objects leave the zone in question

### 6.11.3 Counter

Counters can be configured to count the number of times a rule is triggered. For example, the number of people crossing a line. The counter rule is designed to be utilised in two ways:

- **Increment / Decrement**: whereby a counter is incremented by the attached rule(s) (+1 for each rule trigger), and decremented by another attached rule(s) (-1 for each rule trigger).
- **Occupancy**: whereby the counter reflects the number of objects that are currently triggering the attached rule(s).

More than one rule can be assigned to any of a counter's three inputs. This allows, for example, the occupancy of two presence rules to be reflected in a single counter, or more than one entrance / exit gate to reflect in a single counter. An example rule graph is provided to illustrate this below.

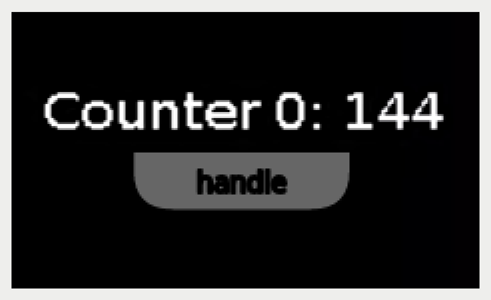
Broadly speaking a single counter should not be used for both purposes occupancy and increment / decrement.

The Counter's Threshold Operator allows the user to limit when a counter generates an event. Based on the selected behaviour and a defined Threshold Value, the counter can be configured to only send events in specific scenarios. Threshold Operators include:

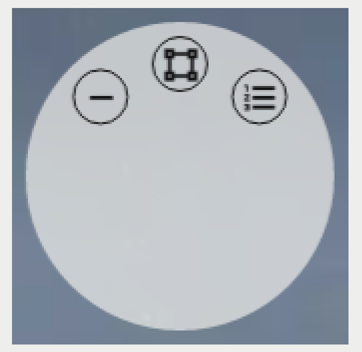
- Greater than or equal to
- Less than or equal to
- Greater than
- Less than
- Equal to
- Not Equal to
- None

#### *6.11.3.1 Positioning Counters*

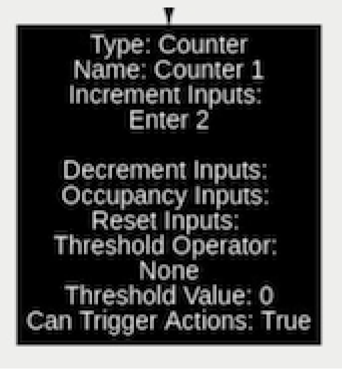
When added, a counter object is visualised on the video stream as seen below. The counter can be repositioned by grabbing the 'handle' beneath the counter name and moving the counter to the desired location.



Right-clicking the mouse (or tap-and-hold on a tablet) on the grid displays the context menu.



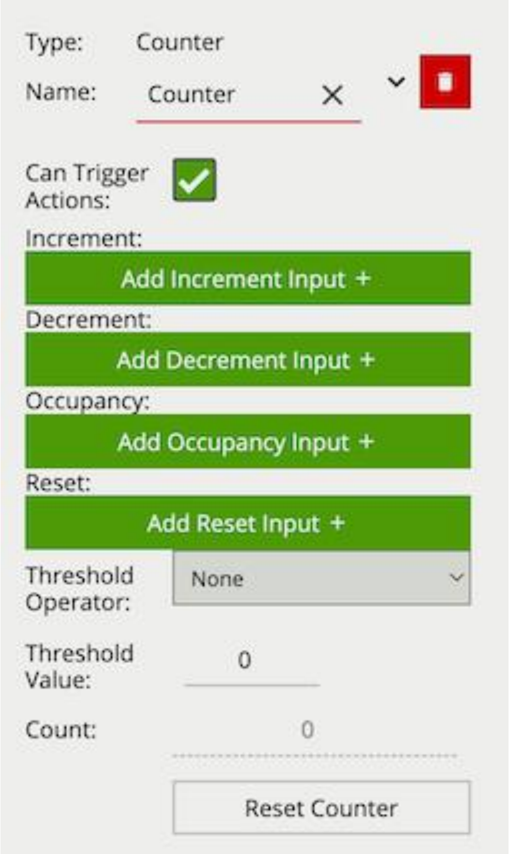
### 6.11.3.2 Graphical View



Type: Counter
Name: Counter 1
Increment Inputs:
Enter 2

Decrement Inputs:
Occupancy Inputs:
Reset Inputs:
Threshold Operator:
None
Threshold Value: 0
Can Trigger Actions: True

### 6.11.3.3 Form View



Type: Counter

Name: Counter

Can Trigger Actions: ☑

Increment:

Decrement:

Occupancy:

Reset:

Threshold Operator: None

Threshold Value: 0

Count: 0

### 6.11.3.4 Configuration

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Counter #” |
| Increment | The rule which, when triggered, will add one to the counter | None |
| Decrement | The rule which, when triggered, will subtract one from the counter | None |
| Occupancy | Sets counter to current number of the rule’s active triggers | None |
| Reset | Resets the count to 0 when the assigned rule(s) trigger | None |

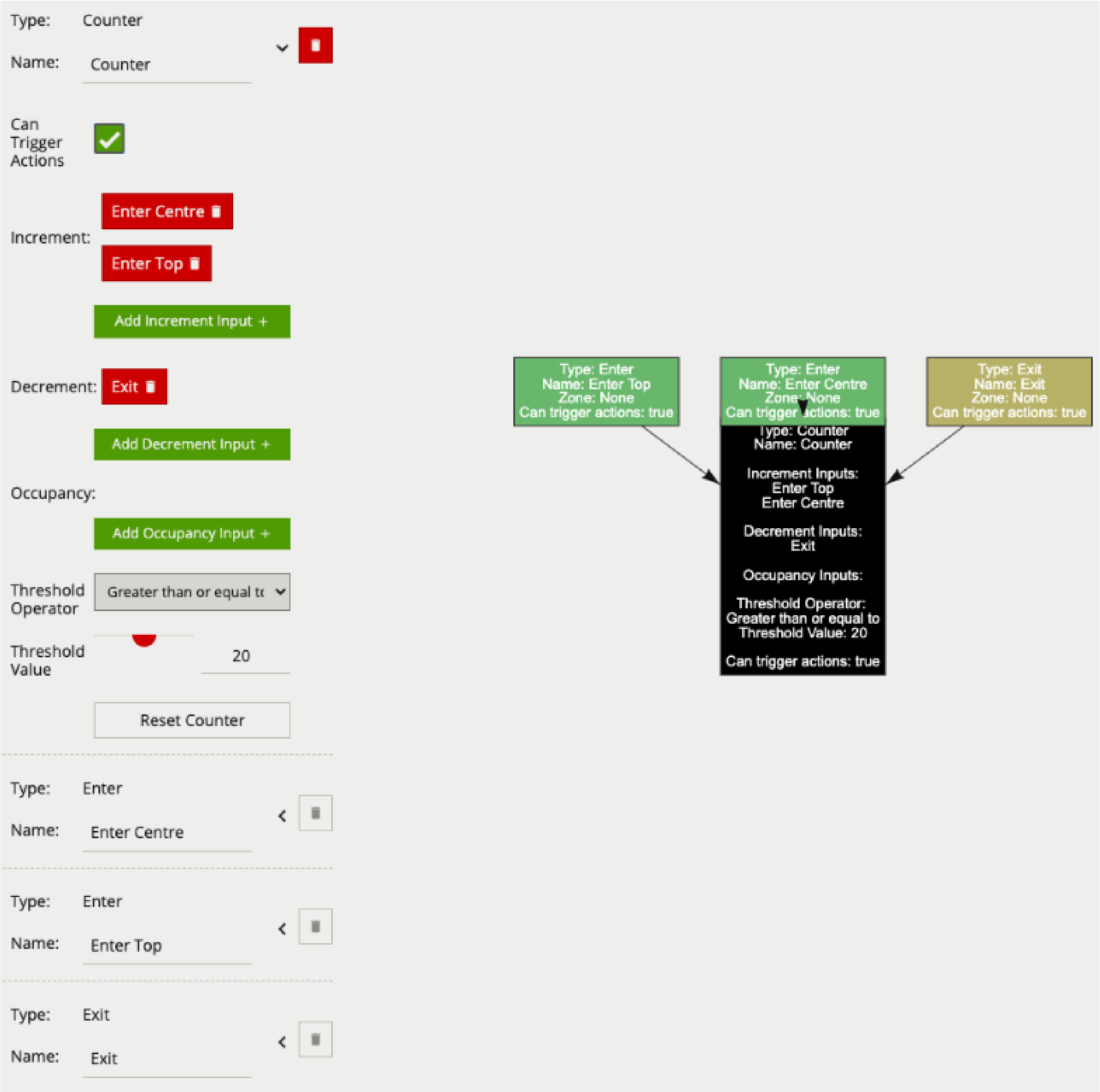
| Property | Description | Default Value |
|---|---|---|
| Threshold Operator | Defines when a Counter will trigger events based on the threshold | None |
| Threshold Value | The value used by the Threshold Operator to define the behaviour | 0 |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Reset Counter | A button allowing the counter value to be reset to 0 | None |

> E.g. if a Presence rule is set as the occupancy target and two objects are currently triggering that Presence rule, the counter will show the value of 2.

#### *6.11.3.5 Typical Logical Rule Combination*

The below counter example increments a counter based on two enter rules, **Enter Centre** and **Enter Top** attached to the zones **Centre** and **Top** respectively, this means that when either of these enter rules triggers the counter will be incremented by + 1. The counter also decrements based on the exit rule **Exit**, which will subtract 1 from the counter each time an object exits the zone **Centre**. The Threshold Operator and Threshold Value, limit the counter to only generate events when the count is more than 20.

Only the counter rule **Counter** is set to **Can Trigger Actions**, meaning only this component of the logical rule will be available as a source for actions. In this case an action using this rule as a source will trigger every time the counter changes.



Type: Counter

Name: Counter

Can Trigger Actions ☑

Increment: Enter Centre, Enter Top

Decrement: Exit

Occupancy:

Threshold Operator: Greater than or equal tc

Threshold Value: 20

Type: Enter

Name: Enter Centre

Type: Enter

Name: Enter Top

Type: Exit

Name: Exit

Type: Enter
Name: Enter Top
Zone: None
Can trigger actions: true

Type: Enter
Name: Enter Centre
Zone: None
Can trigger actions: true

Type: Exit
Name: Exit
Zone: None
Can trigger actions: true

Type: Counter
Name: Counter

Increment Inputs:
Enter Top
Enter Centre

Decrement Inputs:
Exit

Occupancy Inputs:

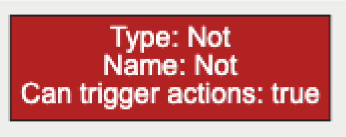
Threshold Operator:
Greater than or equal to
Threshold Value: 20

Can trigger actions: true

### 6.11.4 Not

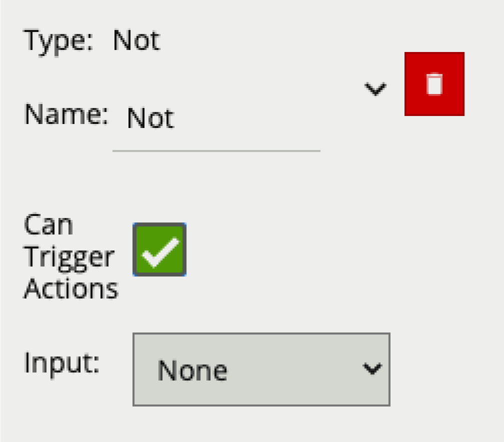
A logical operator that generates an event when the input rule becomes false.

#### 6.11.4.1 Graphical View



Type: Not
Name: Not
Can trigger actions: true

#### 6.11.4.2 Form View



Type: Not

Name: Not

Can Trigger Actions ☑

Input: None

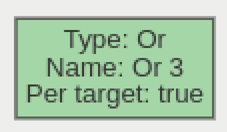
#### 6.11.4.3 Configuration

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Not #” |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |

### 6.11.5 Or

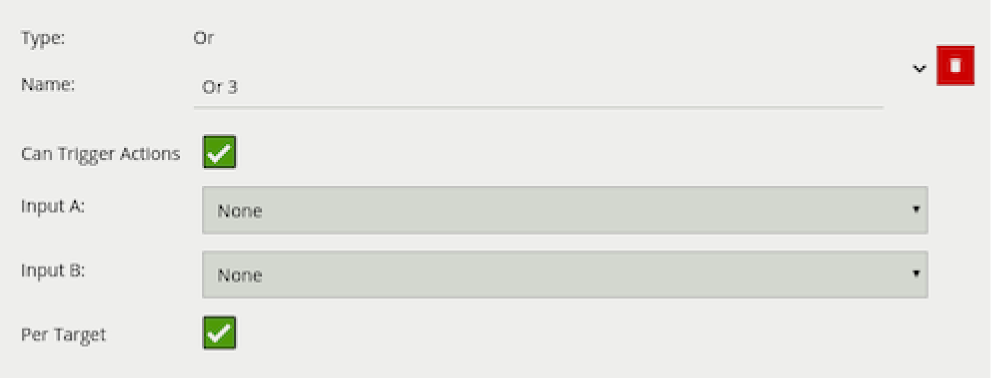
A logical operator that combines two rules and fires events if either input is true.

#### 6.11.5.1 Graphical View



Type: Or
Name: Or 3
Per target: true

### 6.11.5.2 Form View



| | |
|---|---|
| Type: | Or |
| Name: | Or 3 |
| Can Trigger Actions | ☑ |
| Input A: | None |
| Input B: | None |
| Per Target | ☑ |

### 6.11.5.3 Configuration

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | “Not #” |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input A | The first input | None |
| Input B | The second input | None |
| Per Target | Fire one event per tracked object | Active |

If we consider a scene with two Presence rules connected to two separate zones, connected by an OR rule, the table below explains the behaviour of the **Per Target** property.

| State | Per Target | Outcome |
|---|---|---|
| Object A in Input A, No object in input B | On | Two events generated, one for each object |
| No object in Input A, Object B in input B | On | Only one event generated (for Object B) |
| Object A in Input A, No object in input B | On | Only one event generated (for Object A) |
| Object A in Input A, No object in input B | Off | Only one event generated |
| No object in Input A, Object B in input B | Off | Only one event generated |

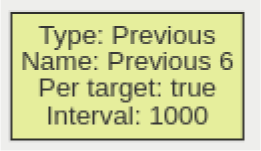
| State | Per Target | Outcome |
|---|---|---|
| Object A in Input A, No object in input B | Off | Only one event generated |

Additionally, it is important to note that if the rule fires when **Per Target** is switched off, it will not fire again until it is 'reset', i.e. until the OR condition is no longer true.

### 6.11.6 Previous

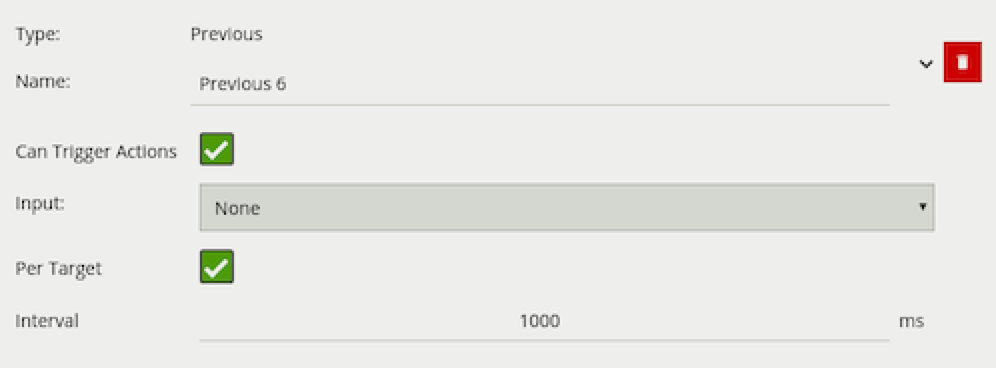
A logical operator that triggers for input events which were active at some point in a past window of time. This window is defined as between the current time and the period before the current time (specified by the Interval value).

#### *6.11.6.1 Graphical View*



Type: Previous
Name: Previous 6
Per target: true
Interval: 1000

#### *6.11.6.2 Form View*



Type: Previous

Name: Previous 6

Can Trigger Actions ☑

Input: None

Per Target ☑

Interval 1000 ms

#### *6.11.6.3 Configuration*

| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Previous #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |

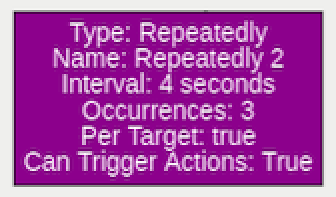
| Property | Description | Default Value |
|---|---|---|
| Input | The input rule | None |
| Per Target | Fire one event per tracked object. See description below for more details | Active |
| Interval | The time in milliseconds | 1 |

### 6.11.7 Repeatedly

A logical operator that triggers when an input rule is triggered a set number of times within a defined period. The Duration period is a window of time computed from every input event. For example, with a Repeatedly rule configured to generate an event when the input triggers three times in eight seconds, and that input rule triggers four times in eight seconds, the repeatedly rule will trigger after both the third input rule trigger and again after the fourth. This is because the first three triggers (events 1-3) fired within an 8 second window, additionally the second set (events 2-4) also occurred within their own 8 second window.

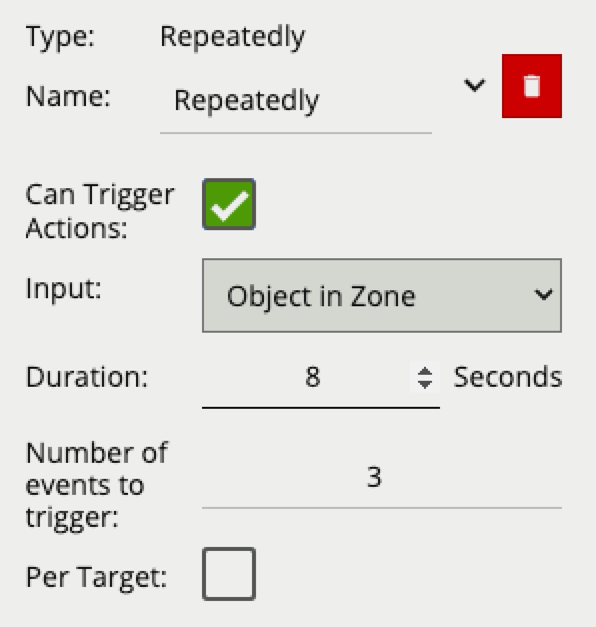
The Per Target option specifies that it must be the same tracked object that triggers the input.

#### *6.11.7.1 Graphical View*



Type: Repeatedly
Name: Repeatedly 2
Interval: 4 seconds
Occurrences: 3
Per Target: true
Can Trigger Actions: True

### 6.11.7.2 Form View



| | |
|---|---|
| Type: | Repeatedly |
| Name: | Repeatedly |
| Can Trigger Actions: | ☑ |
| Input: | Object in Zone |
| Duration: | 8 Seconds |
| Number of events to trigger: | 3 |
| Per Target: | □ |

### 6.11.7.3 Configuration

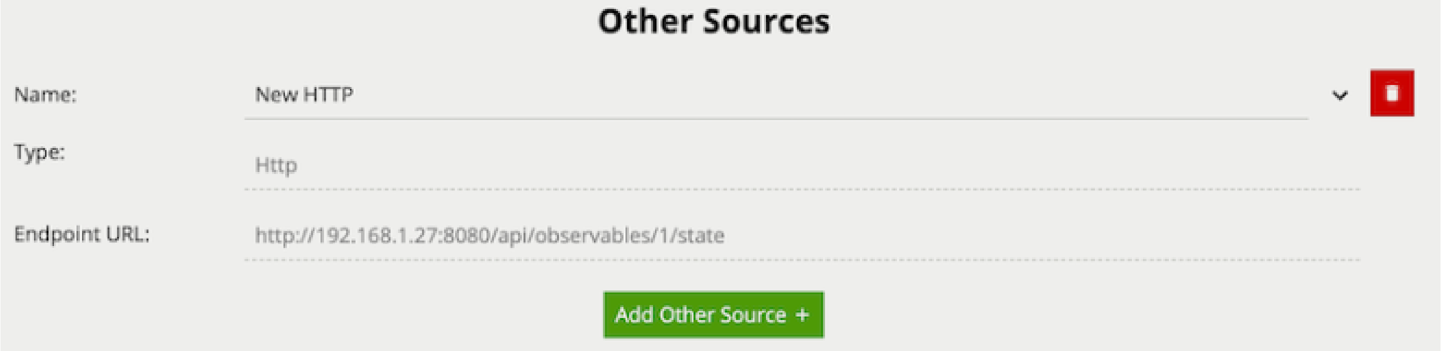
| Property | Description | Default Value |
|---|---|---|
| Name | A user-specified name for this rule | "Repeatedly #" |
| Can Trigger Actions | Specifies whether events generated by this rule trigger actions | Active |
| Input | The input rule | None |
| Duration | The time in which the Number of Events to Trigger must fire | 3 |
| Number of Events to Trigger | The number of times the input is required to trigger | 4 |
| Per Target | Specifies if the input needs to be triggered by the same object | Inactive |

## 6.12 Other Sources

- HTTP
- Schedule

### 6.12.1 HTTP

The HTTP source creates an arbitrary REST API endpoint with a state variable that can be set true or false. This creates a virtual Digital Input which third party systems can enable or disable. The HTTP source can be referenced by the [Source Filter] in a rule graph.



Other Sources

Name: New HTTP

Type: Http

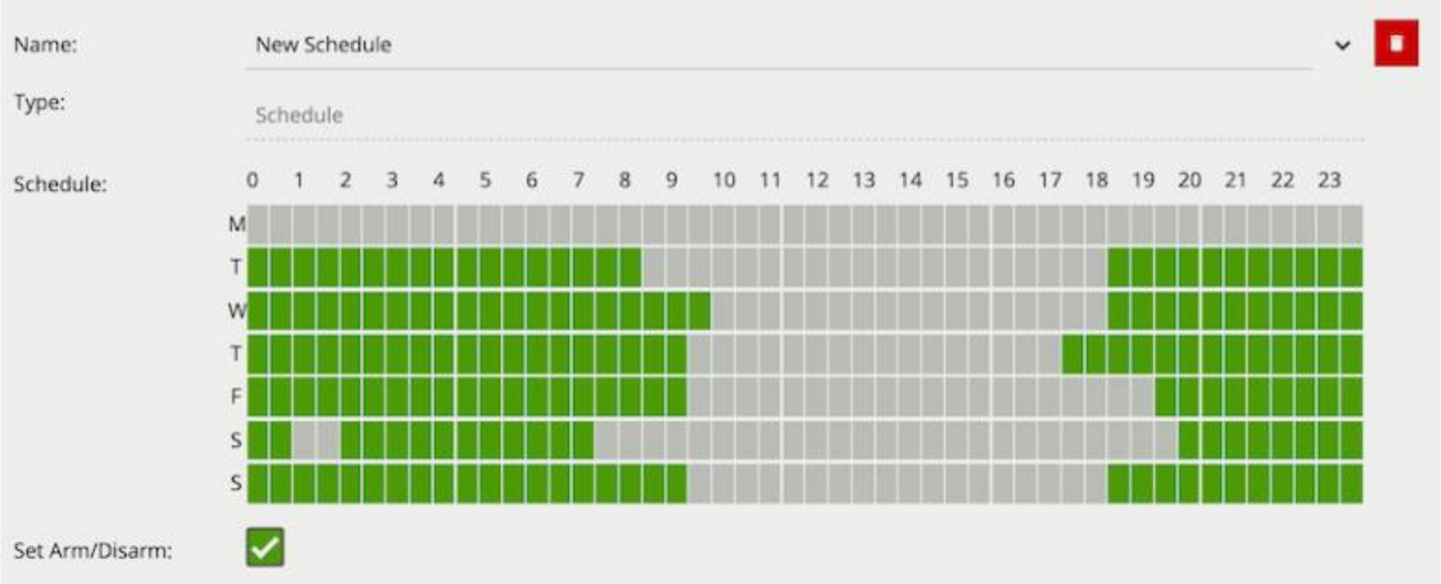
Endpoint URL: http://192.168.1.27:8080/api/observables/1/state

#### *6.12.1.1 Properties*

- **Endpoint URL:** The REST API endpoint defining the state variable.

### 6.12.2 Schedule

The Schedule source allows the definition of a schedule of time when the source is either on or off. The Schedule other source can be referenced by the [Source Filter] in a rule graph. Additionally, the schedule source can be used to directly control the armed state of VCA.



Name: New Schedule

Type: Schedule

Schedule:

| | 0 | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | 18 | | 19 | | 20 | | 21 | | 22 | | 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ |
| T | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| W | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| T | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| F | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| S | 🟩 | 🟩 | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| S | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |

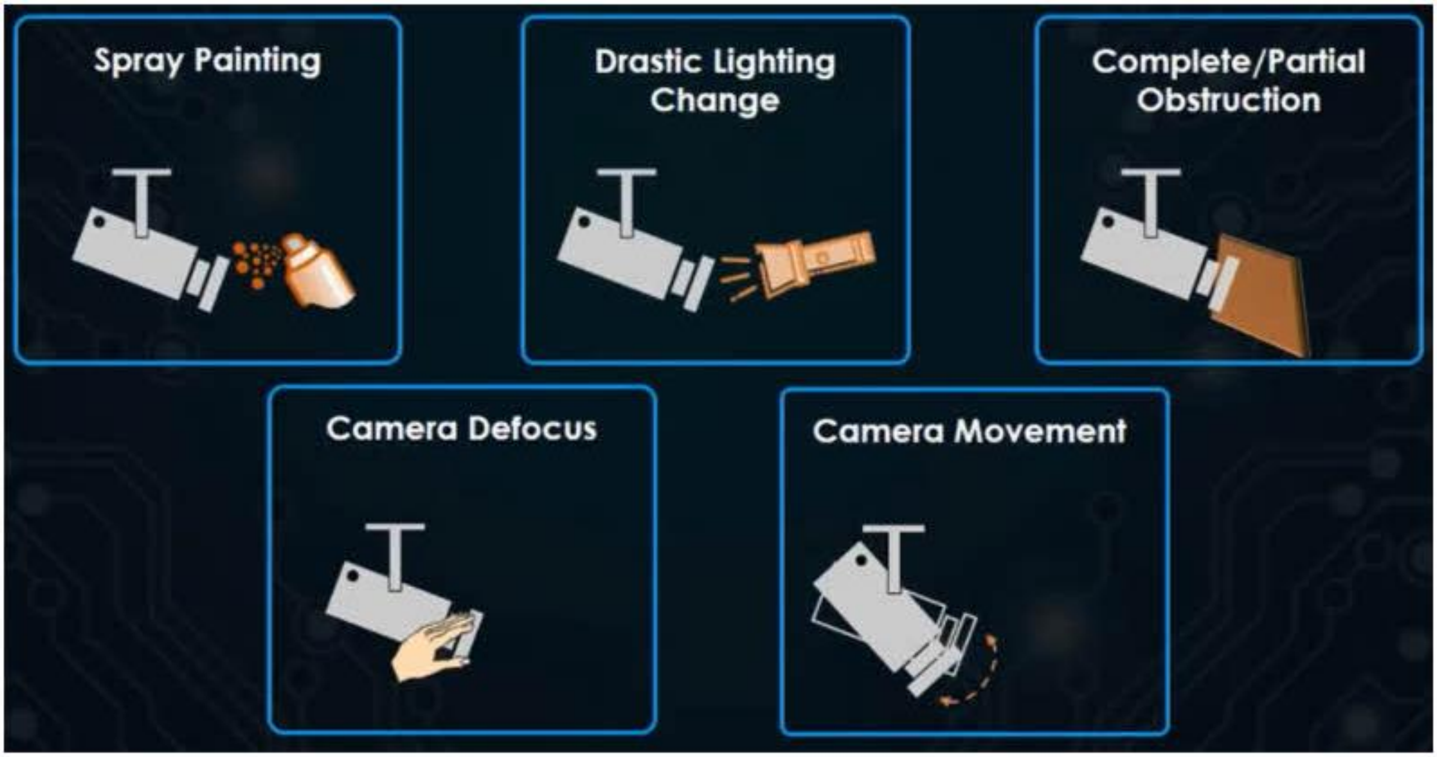
Set Arm/Disarm: ☑

#### *6.12.2.1 Properties*

- **Schedule:** A click and drag interface which allows the definition of on periods (in green) and off periods (in grey). Each row represents one of the seven days in a week and each column represents a half hour period in that 24 hours.

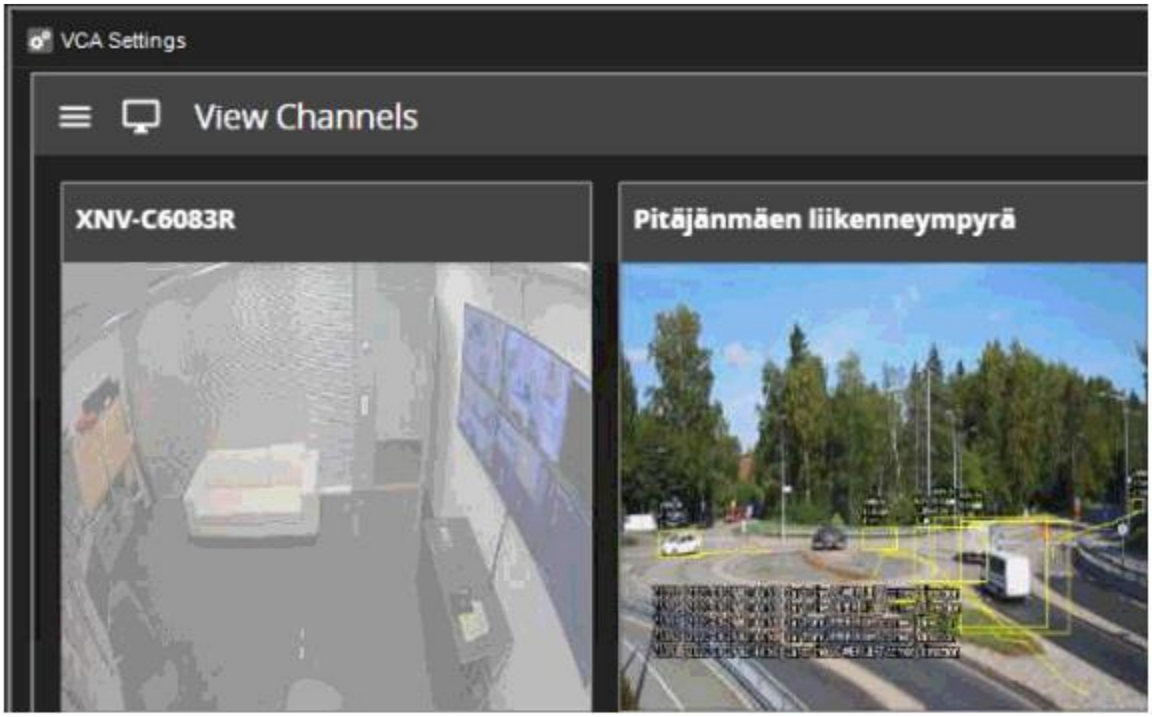
- **Set Arm/Disarm:** When checked, the schedule source directly sets VCA Armed state according to the schedule defined above.

## 6.13 VCA – Tamper Detection

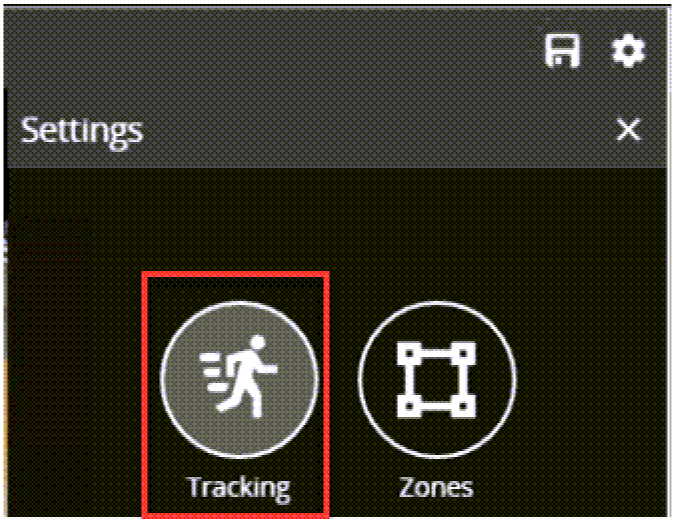


### 6.13.1 How to enable Tamper Detection

1. Open camera from the View Channels



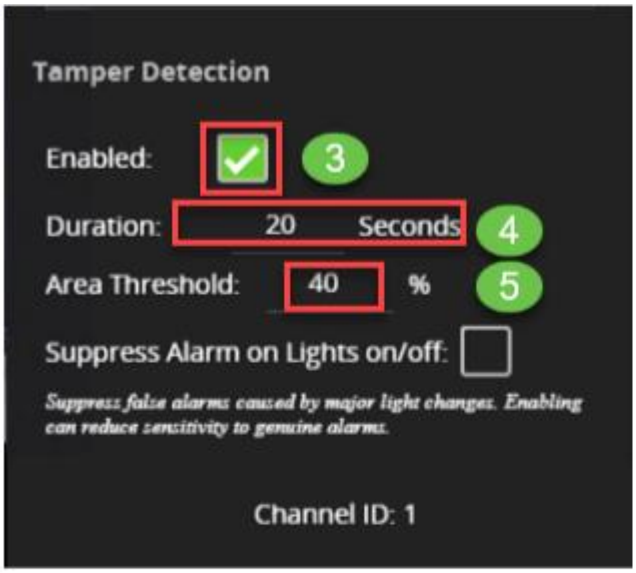
2. Open **Tracking**



3. Enable Tamper Detection

4. Set **Duration**

5. Set **Area Threshold**



Tamper Detection

Enabled: ☑ 3

Duration: 20 Seconds 4

Area Threshold: 40 % 5

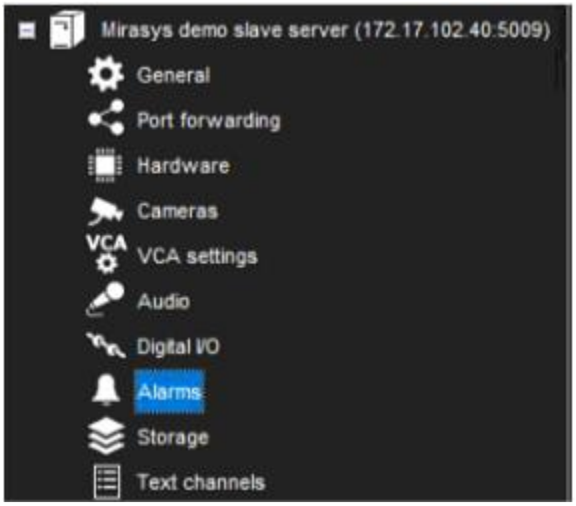
Suppress Alarm on Lights on/off: □

*Suppress false alarms caused by major light changes. Enabling can reduce sensitivity to genuine alarms.*

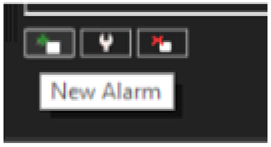
Channel ID: 1

### 6.13.2 How to create an alarm from Tamper Detection

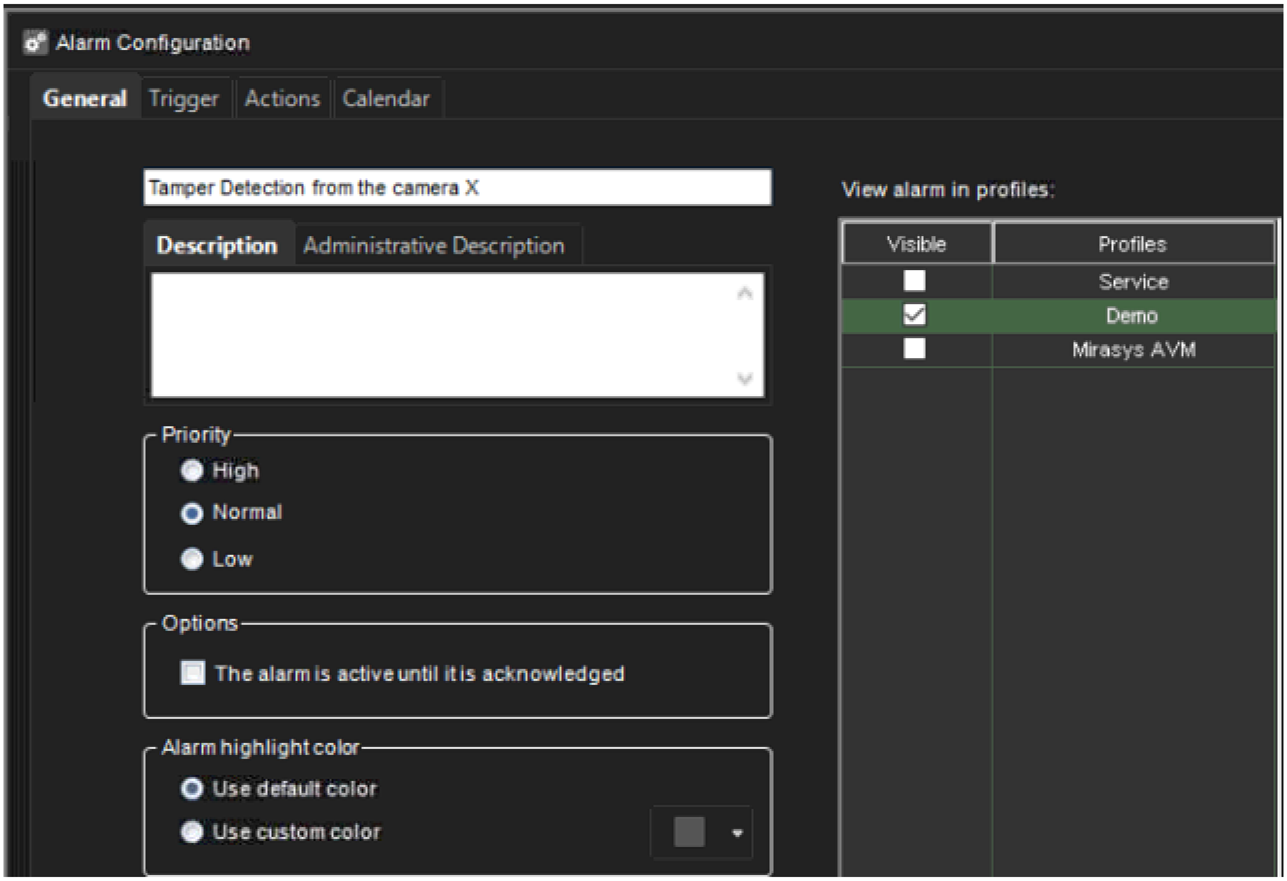
1. Open Alarms from the needed VMS server



2. Click **New Alarm**



3. Enter the name of the alarm


Alarm Configuration

General | Trigger | Actions | Calendar

Tamper Detection from the camera X

Description | Administrative Description

View alarm in profiles:

| Visible | Profiles |
|---|---|
| ☐ | Service |
| ☑ | Demo |
| ☐ | Mirasys AVM |

Priority
- ○ High
- ● Normal
- ○ Low

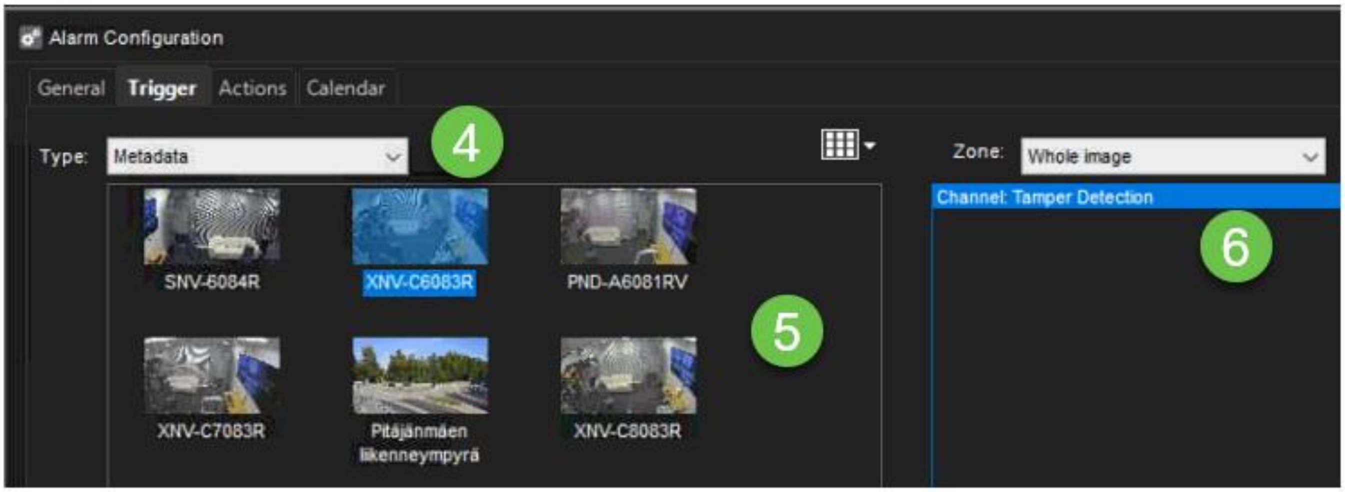
Options
- ☐ The alarm is active until it is acknowledged

Alarm highlight color
- ● Use default color
- ○ Use custom color

4. Select Trigger **Metadata**

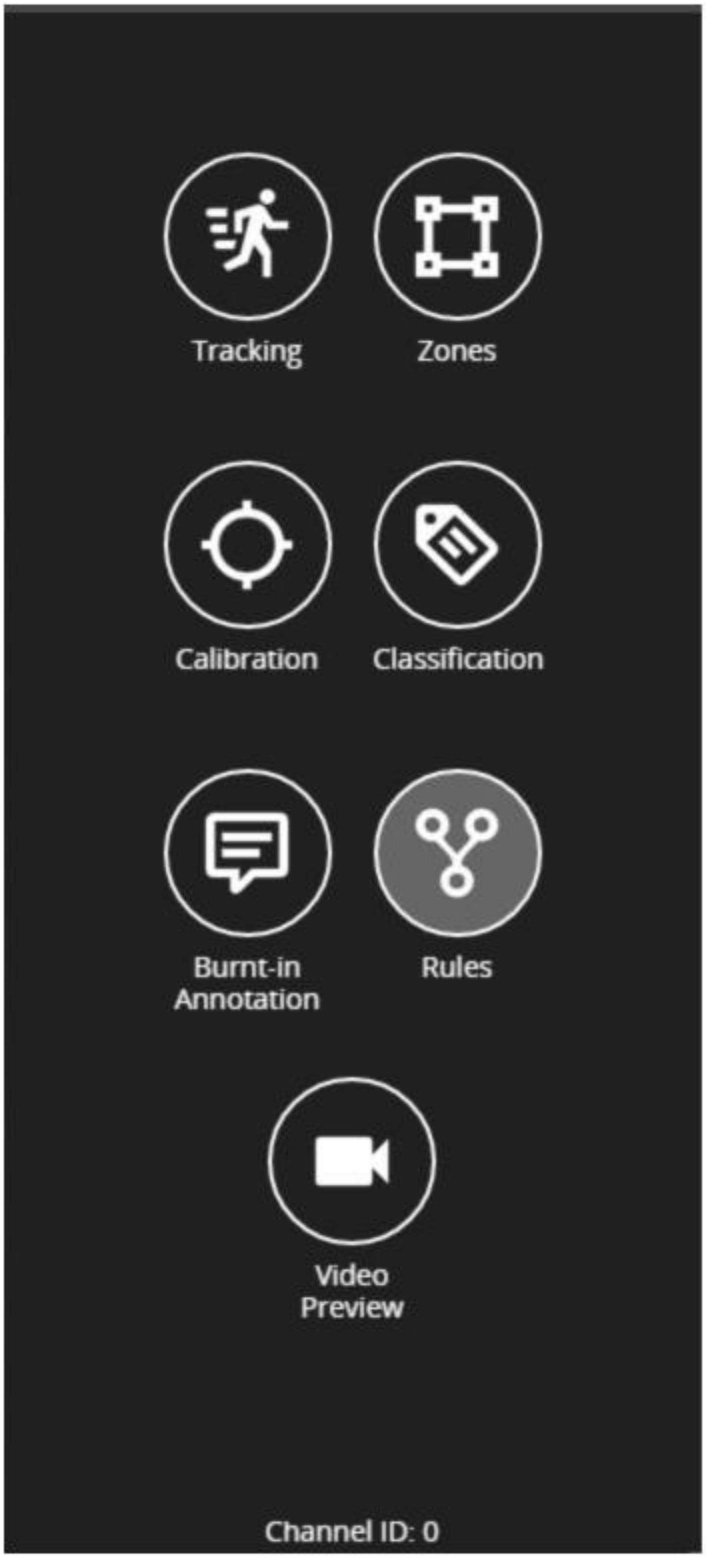
5. Select camera from the list

6. Select Tamper Detection

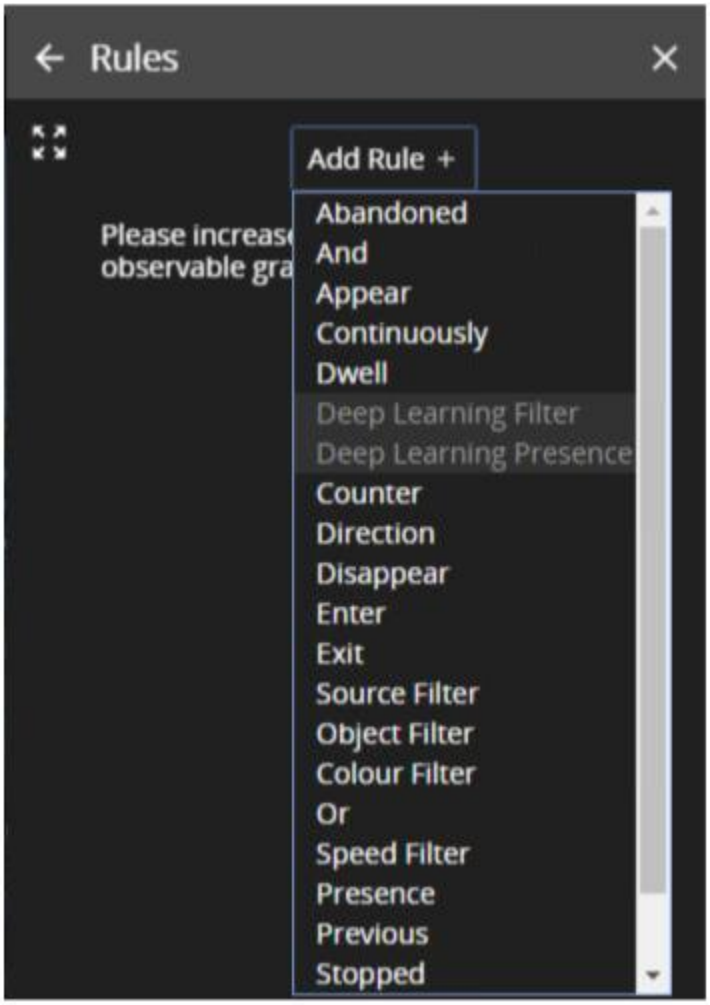


## 6.14 How to create rules

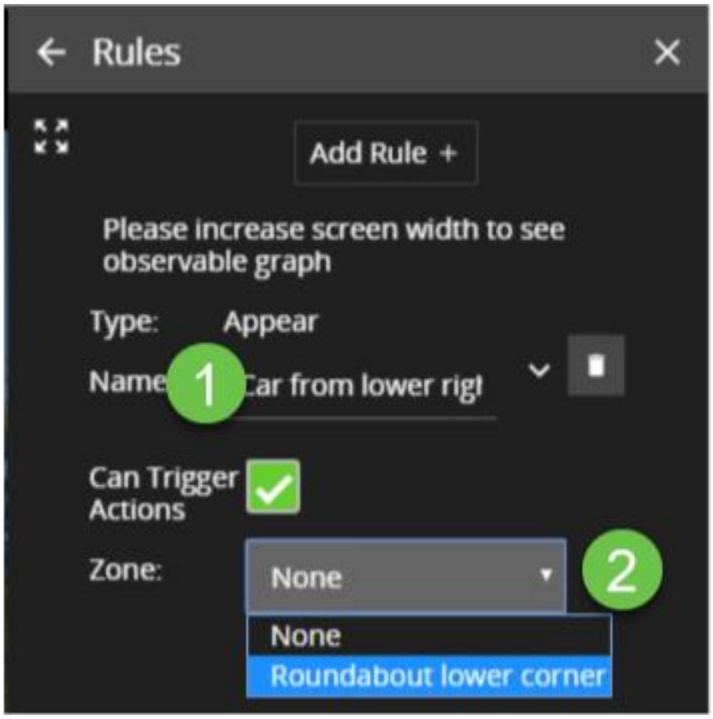
1. Click **Rules**


Tracking
Zones
Calibration
Classification
Burnt-in Annotation
Rules
Video Preview
Channel ID: 0

2. Click **Add Rule**
3. Select rule from the list



4. Set the name for the rule
5. Select zone



Please increase screen width to see observable graph

Type: Appear

Name: Car from lower rigt

Can Trigger Actions ☑

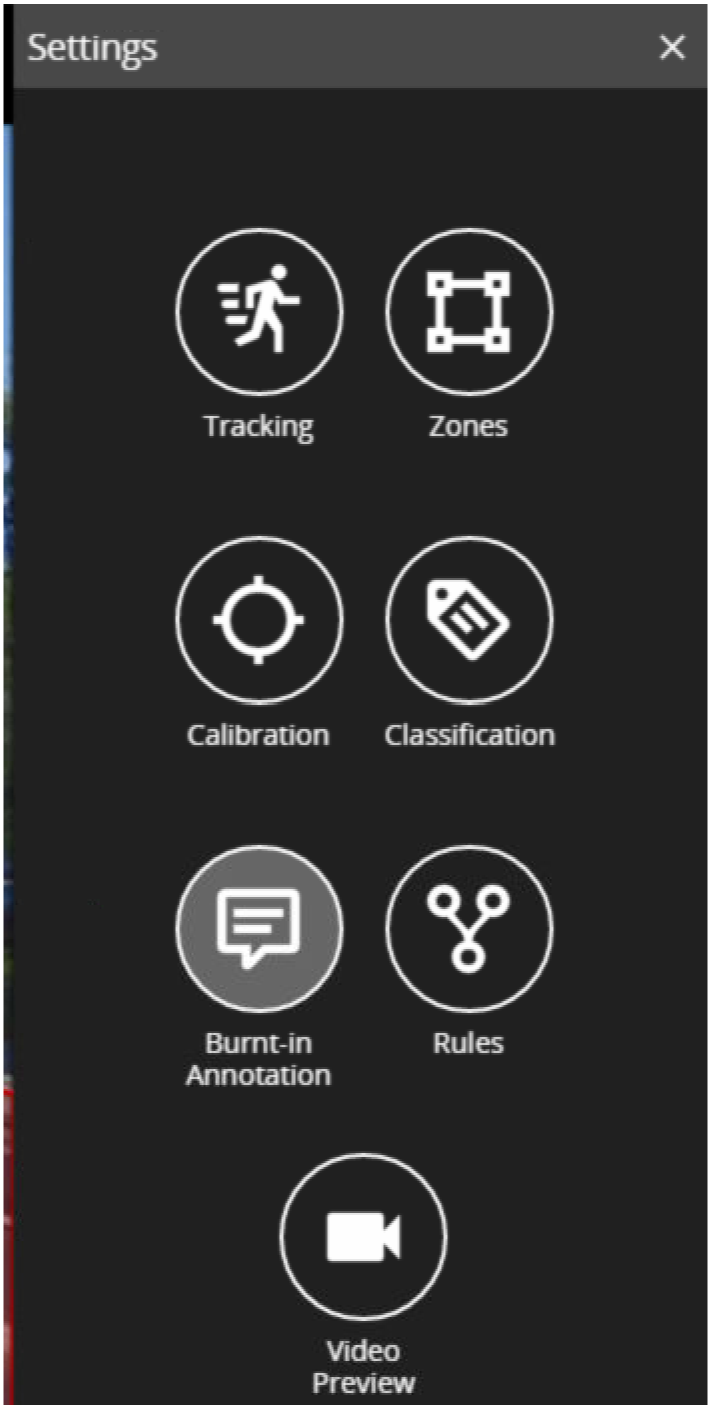
Zone: None

None

Roundabout lower corner

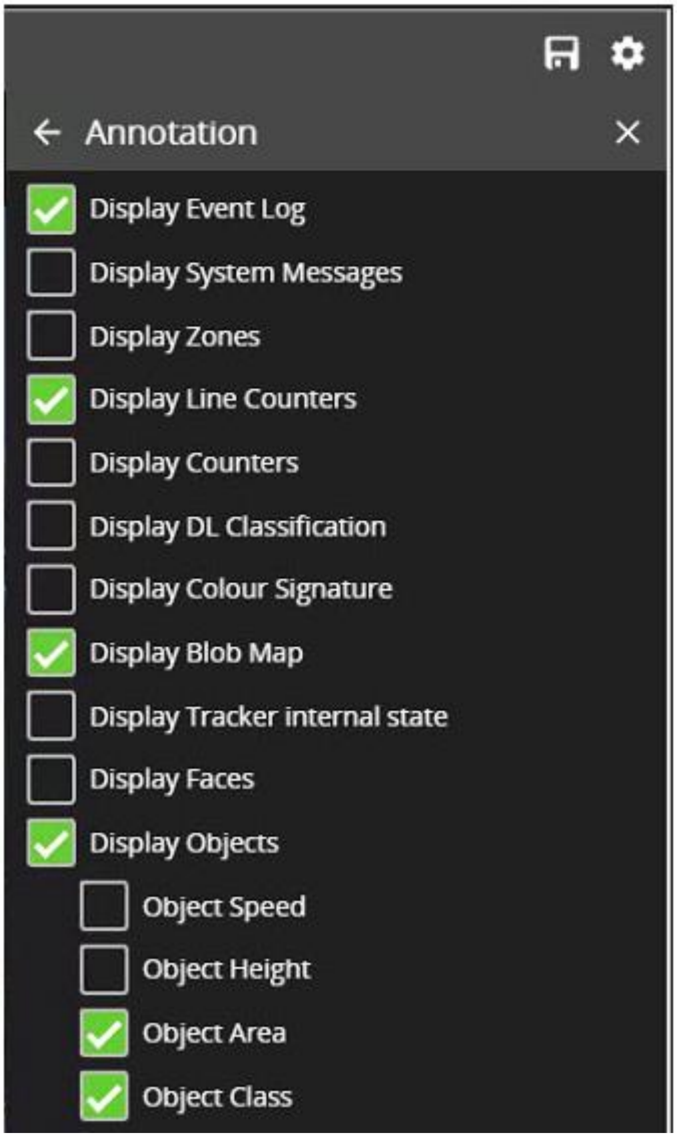
## 6.15 How to test rules

1. Open **Burnt-in Annotation**



2. Enable **Event Log**

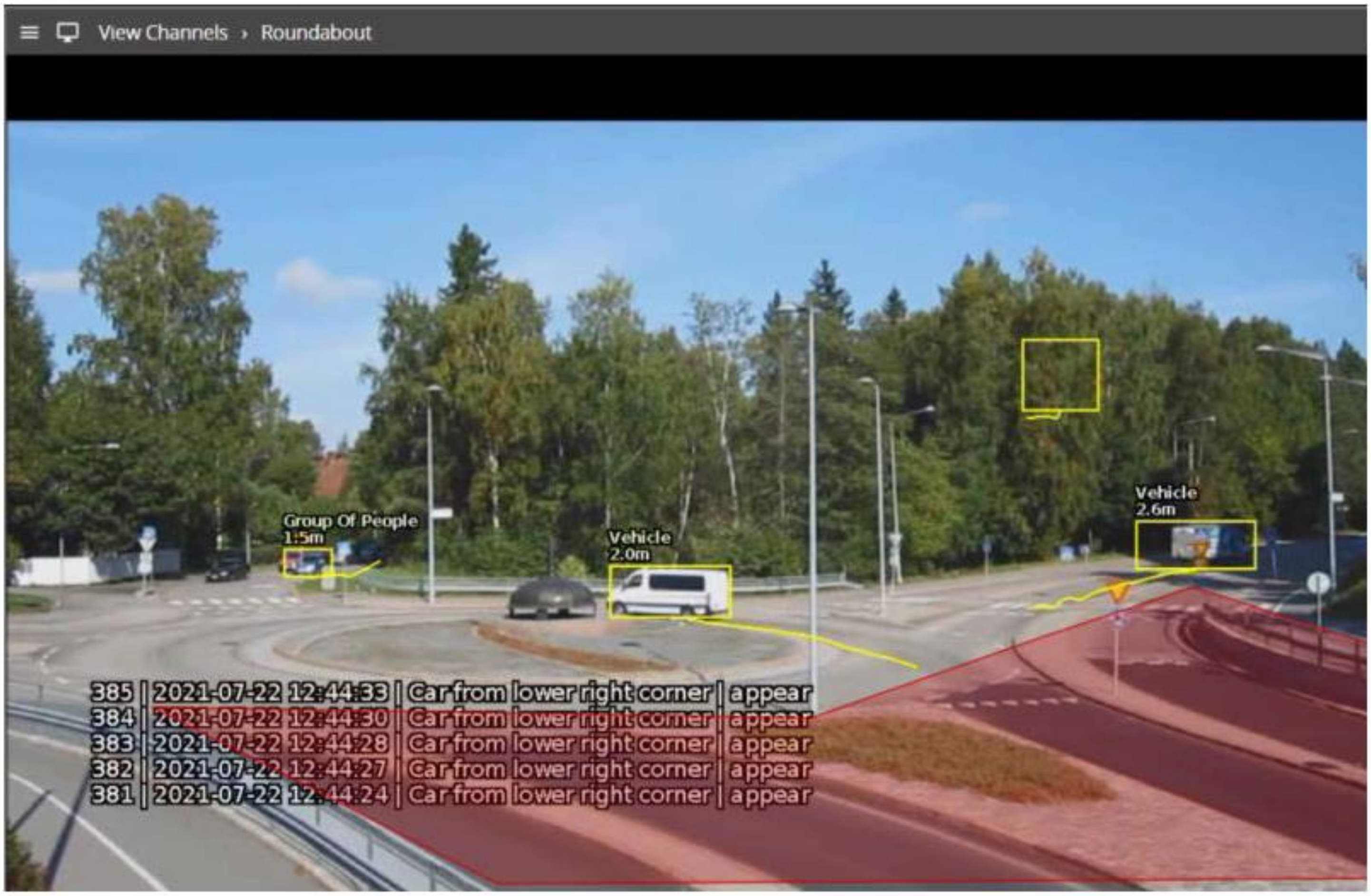
3. Enable **Display Line Counters**
4. Enable **Display Blob Map**
5. Enable **Display Objects**
    a. Enable **Object Area**
    b. Enable **Object Class**



Annotation

☑ Display Event Log
□ Display System Messages
□ Display Zones
☑ Display Line Counters
□ Display Counters
□ Display DL Classification
□ Display Colour Signature
☑ Display Blob Map
□ Display Tracker internal state
□ Display Faces
☑ Display Objects
- □ Object Speed
- □ Object Height
- ☑ Object Area
- ☑ Object Class

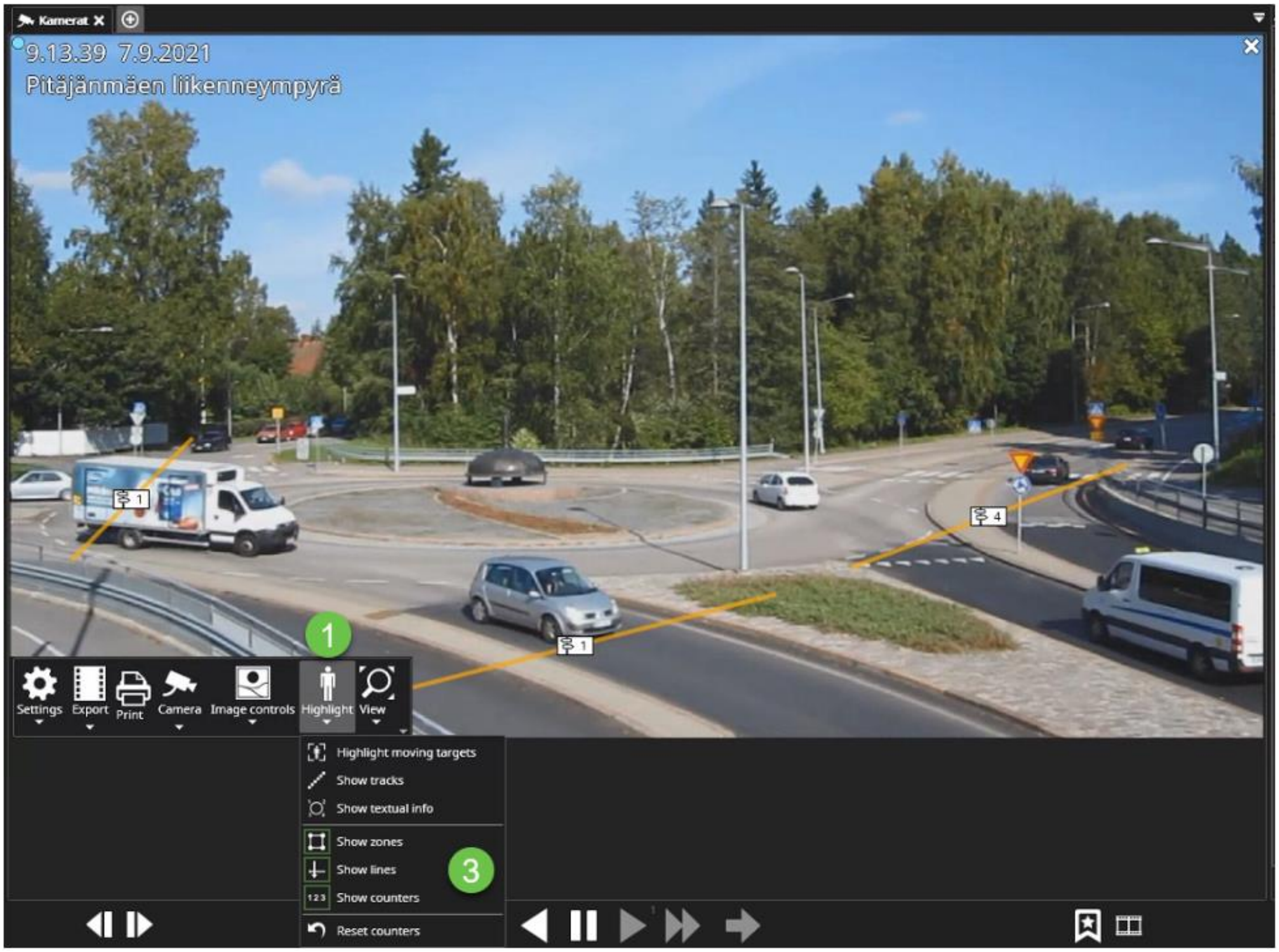
Open the correct camera and check that triggered events can be seen



```
385 | 2021-07-22 12:44:33 | Car from lower right corner | appear
384 | 2021-07-22 12:44:30 | Car from lower right corner | appear
383 | 2021-07-22 12:44:28 | Car from lower right corner | appear
382 | 2021-07-22 12:44:27 | Car from lower right corner | appear
381 | 2021-07-22 12:44:24 | Car from lower right corner | appear
```

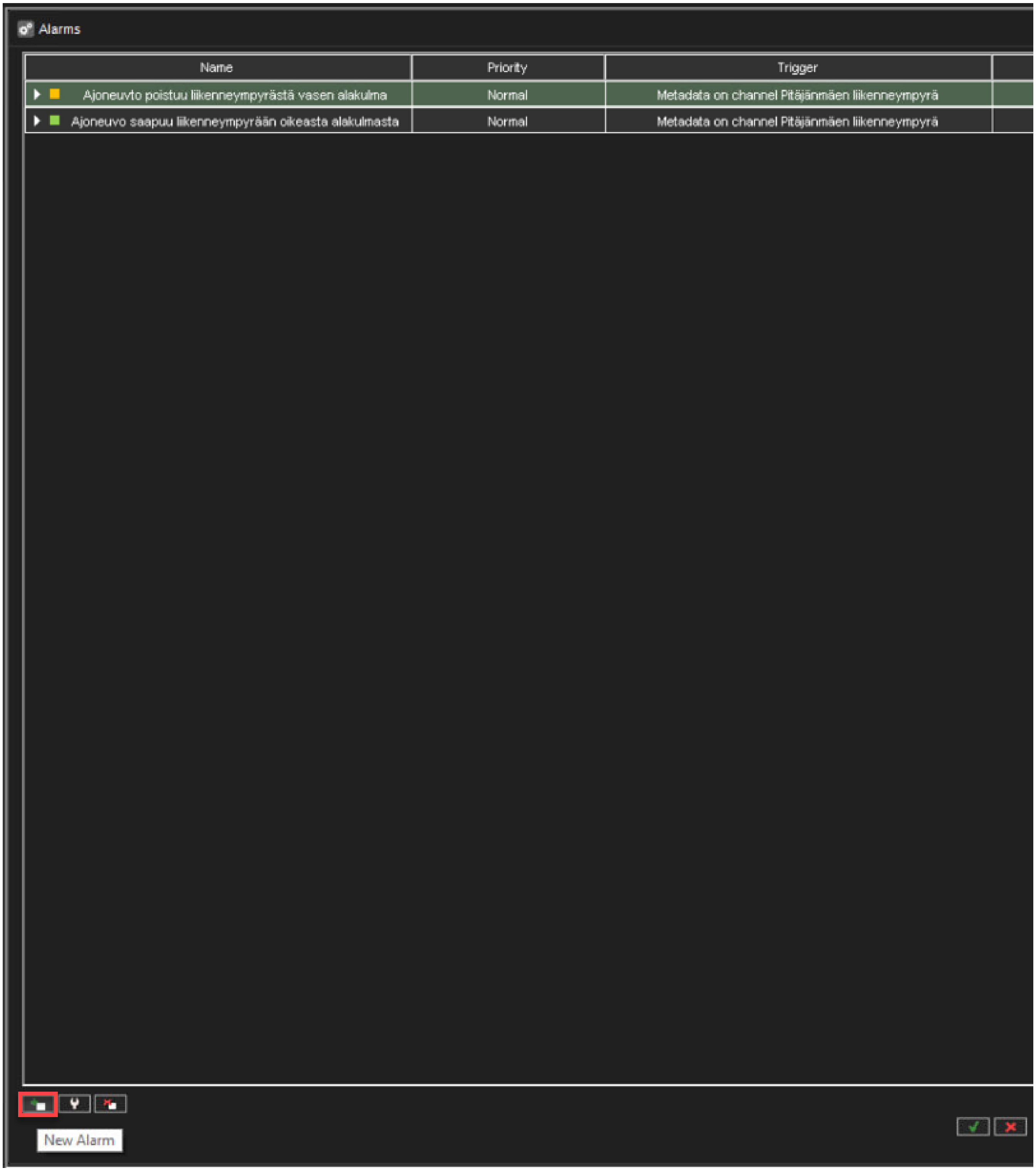
## 6.16 How to view VCA events in the Mirasys Spotter

1. Open camera to the real-time view
2. Open camera toolbar and click **Highlight**
3. Enable needed options(**Show zone**s, **Show lines** or **Show counters**

1. 

## 6.17 How to create an alarm from VCA event

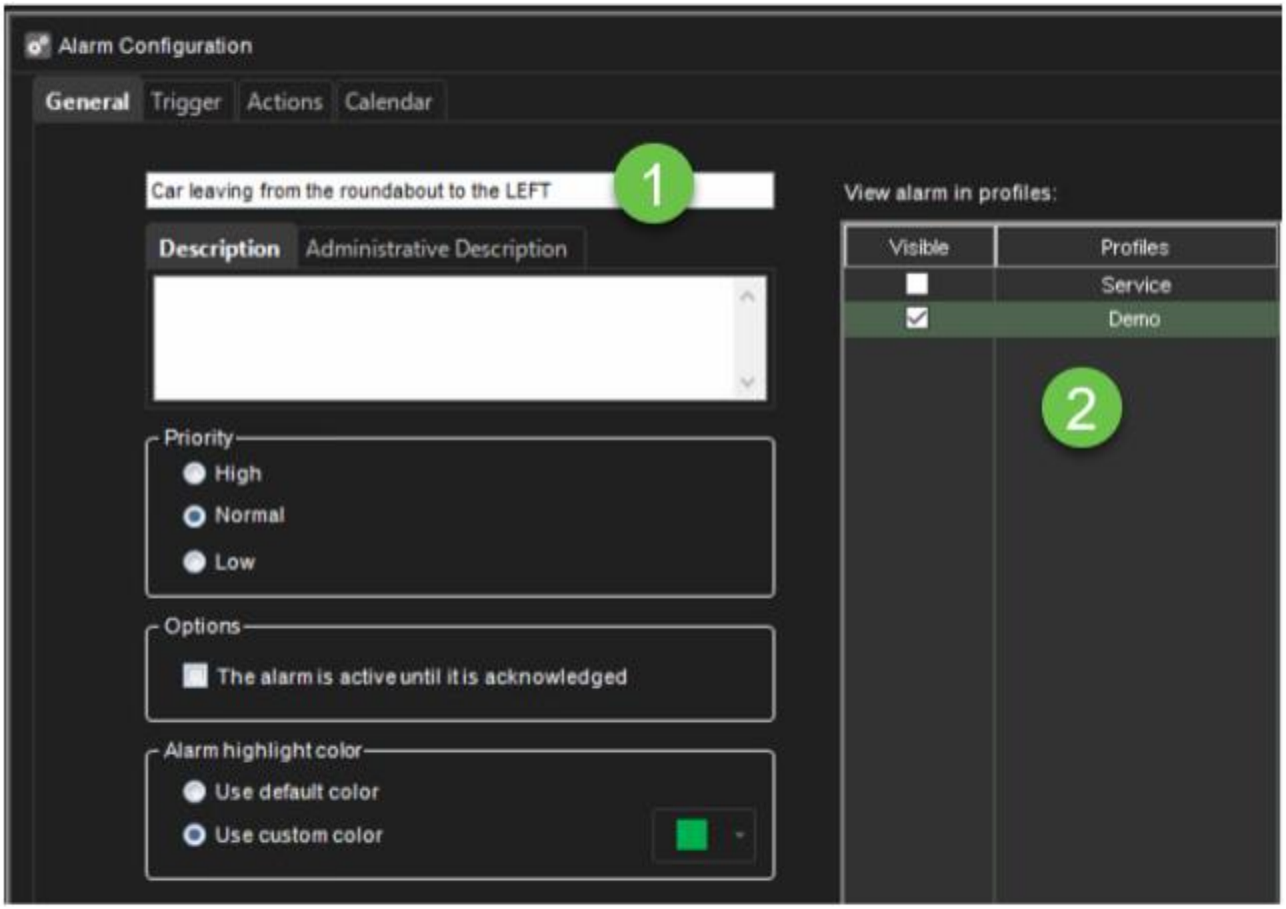
1. Go to the **VMS servers** tab
2. Click **Alarms**
3. Click **New Alarm**



Alarms

| Name | Priority | Trigger |
|---|---|---|
| ▶ 🟧 Ajoneuvto poistuu liikenneympyrästä vasen alakulma | Normal | Metadata on channel Pitäjänmäen liikenneympyrä |
| ▶ 🟩 Ajoneuvo saapuu liikenneympyrään oikeasta alakulmasta | Normal | Metadata on channel Pitäjänmäen liikenneympyrä |

4. Enter the name of the alarm
5. Select the **View alarm in profiles**
6. Go to the **Trigger** tab



Car leaving from the roundabout to the LEFT

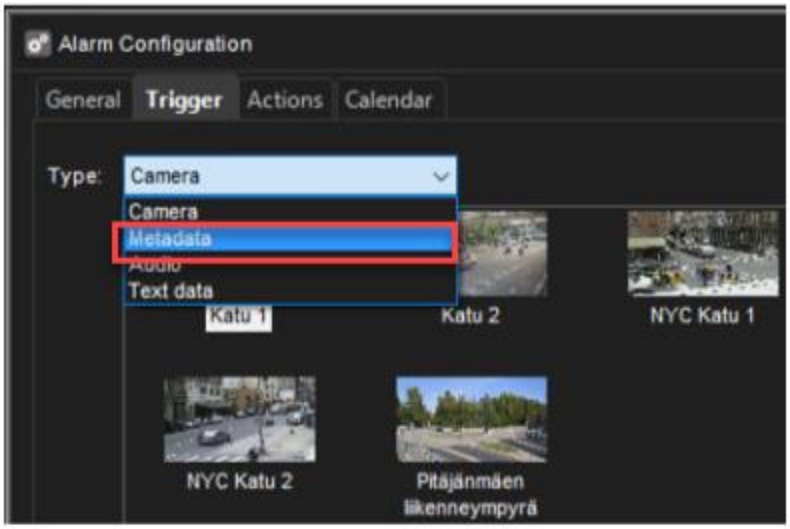
| Visible | Profiles |
|---|---|
| □ | Service |
| ☑ | Demo |

Priority
○ High
● Normal
○ Low

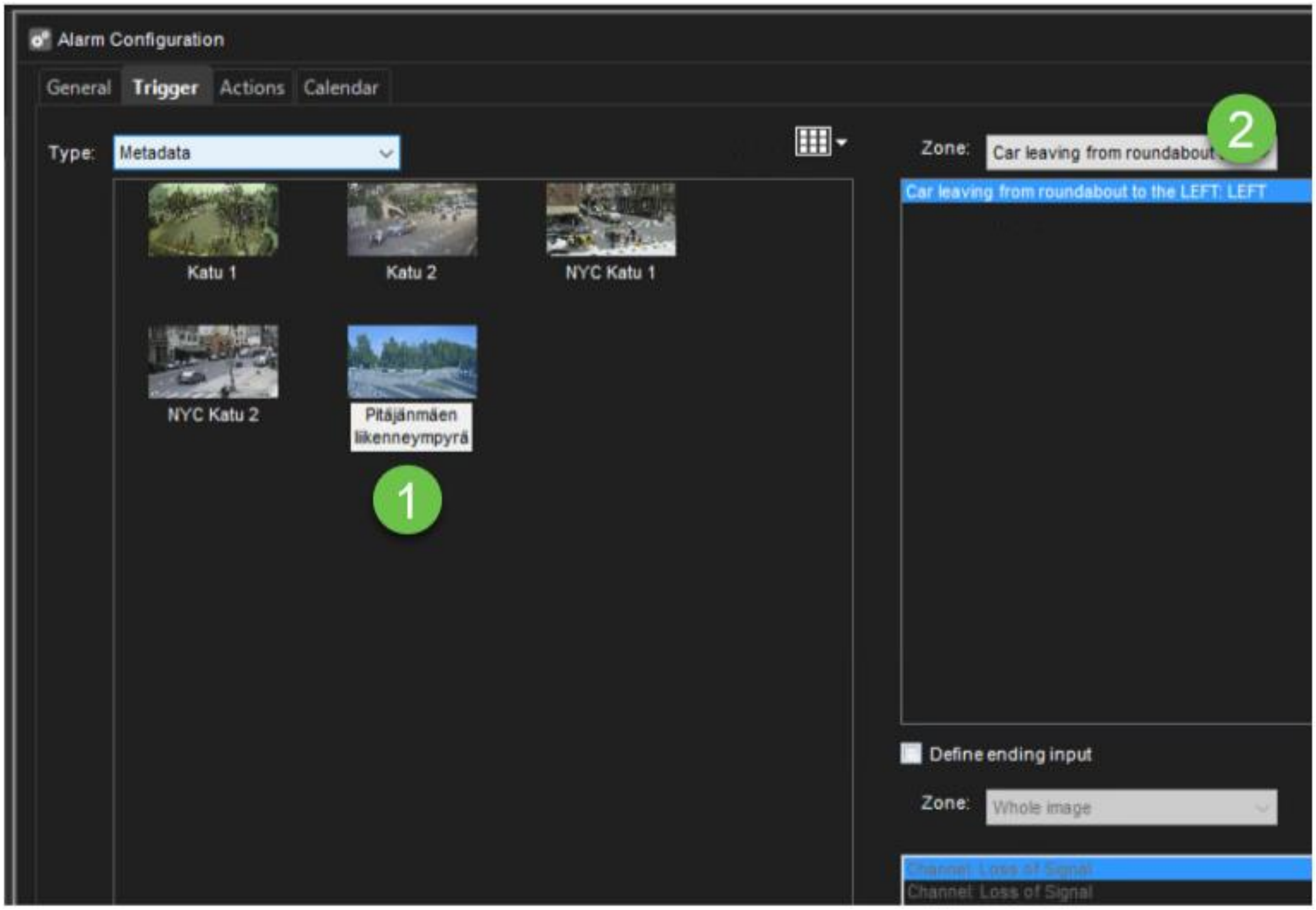
Options
□ The alarm is active until it is acknowledged

Alarm highlight color
○ Use default color
● Use custom color 🟩

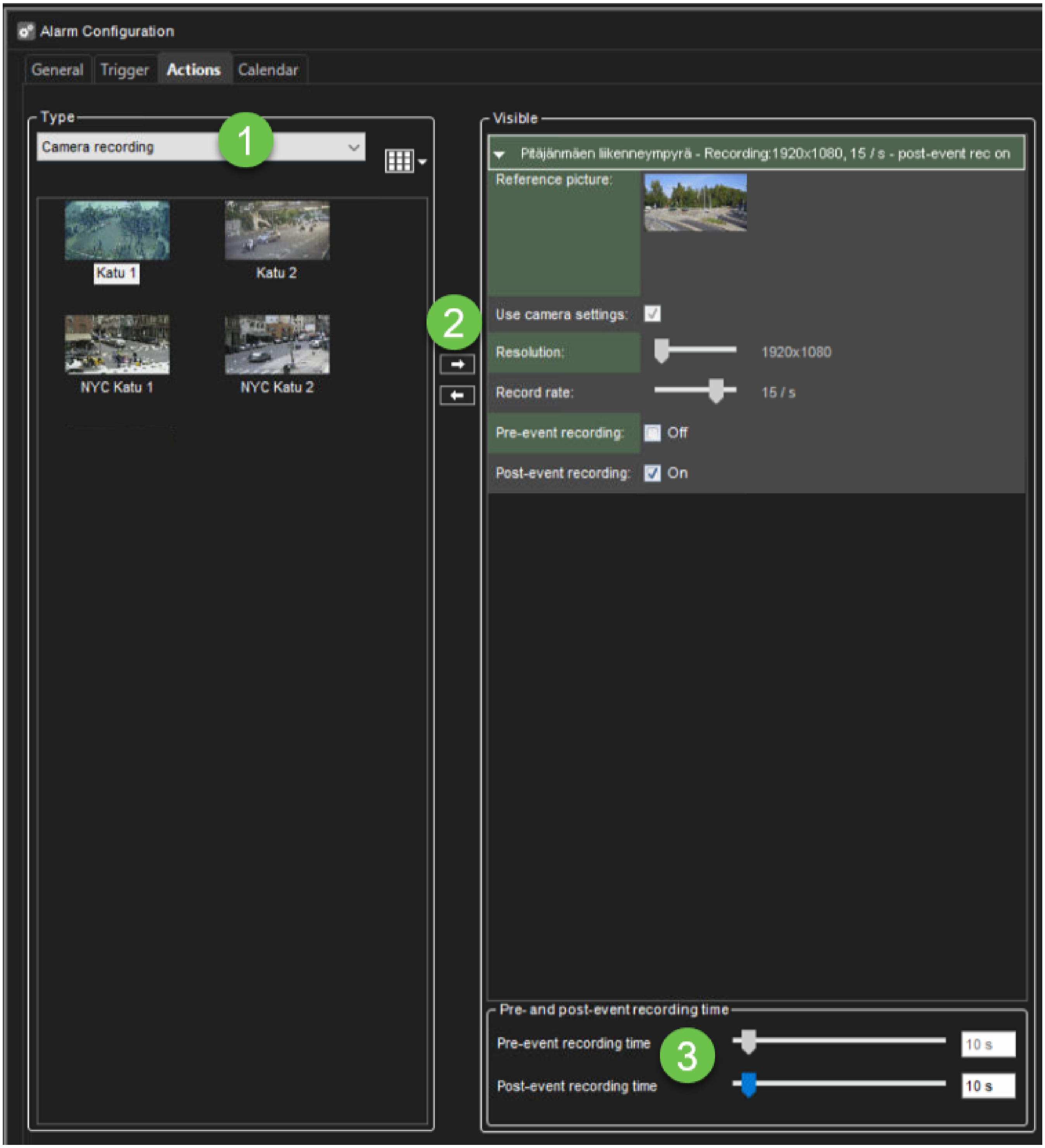
7. Select **Type**: **Metadata**



8. Select the correct camera from the list
9. Select zone
10. Go to the **Actions** tab



11. Select needed actions
12. Add them to the Visible list
13. Set Pre - and post-event recording time, if needed
14. Go to the **Calendar** tab



Type: Camera recording

Visible

Pitäjänmäen liikenneympyrä - Recording:1920x1080, 15 / s - post-event rec on

Reference picture:

Use camera settings: ☑

Resolution: 1920x1080

Record rate: 15 / s

Pre-event recording: □ Off

Post-event recording: ☑ On

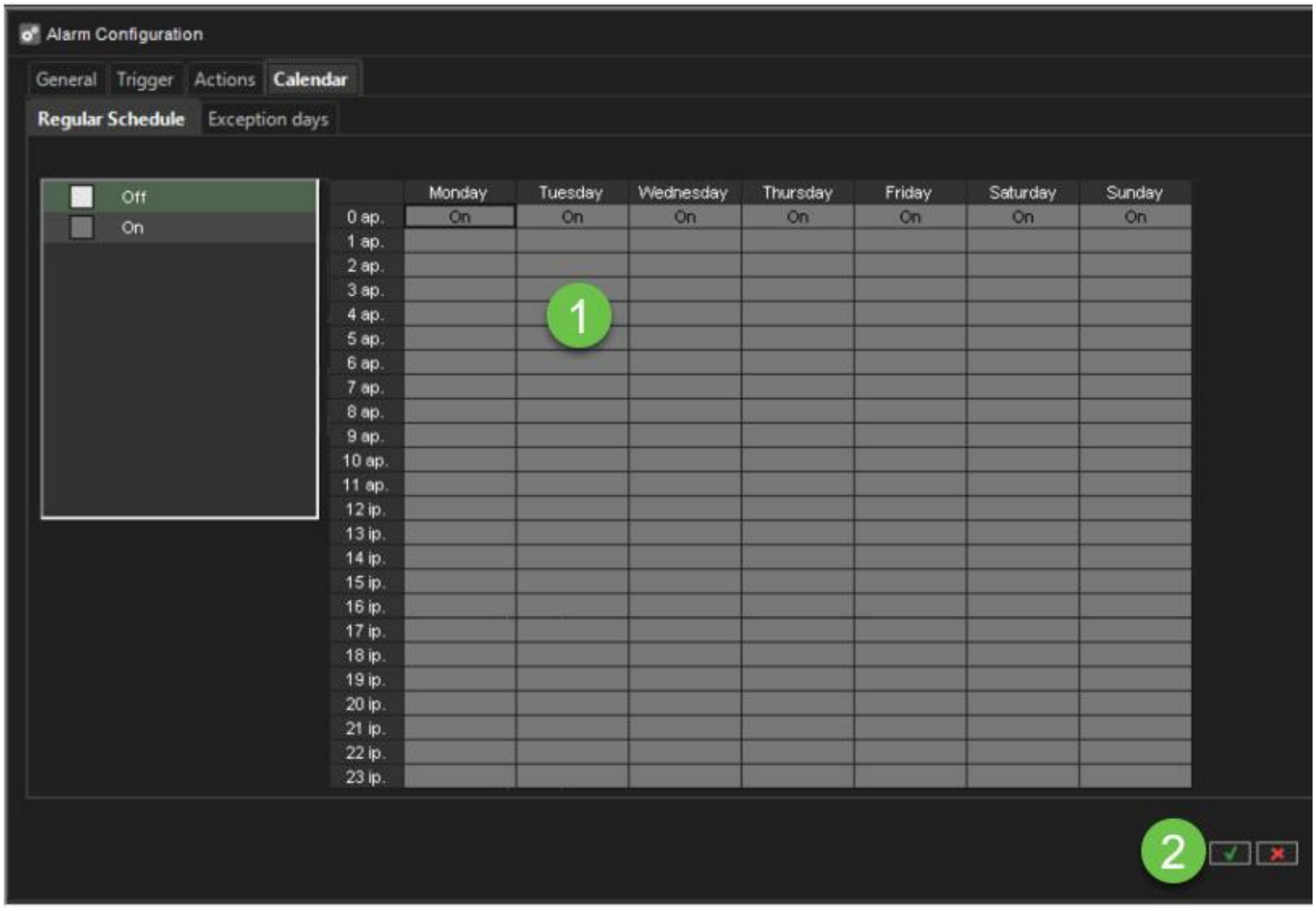
Pre- and post-event recording time

Pre-event recording time: 10 s

Post-event recording time: 10 s

15. Set alarm schedule

16. Click OK

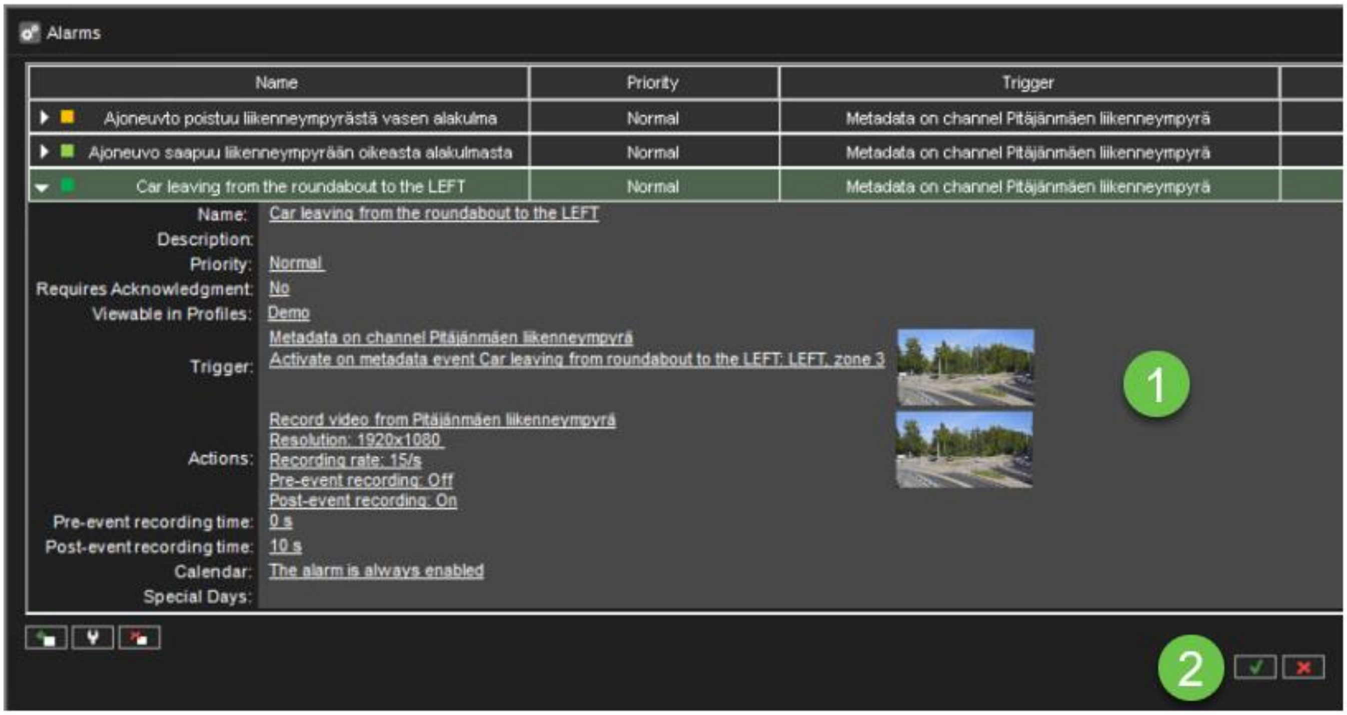


| | |
|---|---|
| □ | Off |
| ■ | On |

| | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|---|---|---|---|---|---|---|---|
| 0 ap. | On | On | On | On | On | On | On |
| 1 ap. | | | | | | | |
| 2 ap. | | | | | | | |
| 3 ap. | | | | | | | |
| 4 ap. | | | | | | | |
| 5 ap. | | | | | | | |
| 6 ap. | | | | | | | |
| 7 ap. | | | | | | | |
| 8 ap. | | | | | | | |
| 9 ap. | | | | | | | |
| 10 ap. | | | | | | | |
| 11 ap. | | | | | | | |
| 12 ip. | | | | | | | |
| 13 ip. | | | | | | | |
| 14 ip. | | | | | | | |
| 15 ip. | | | | | | | |
| 16 ip. | | | | | | | |
| 17 ip. | | | | | | | |
| 18 ip. | | | | | | | |
| 19 ip. | | | | | | | |
| 20 ip. | | | | | | | |
| 21 ip. | | | | | | | |
| 22 ip. | | | | | | | |
| 23 ip. | | | | | | | |

17. Check alarm configuration

18. Click OK to finalize alarm creation



Alarms

| Name | Priority | Trigger |
|---|---|---|
| ▶ 🟨 Ajoneuvto poistuu liikenneympyrästä vasen alakulma | Normal | Metadata on channel Pitäjänmäen liikenneympyrä |
| ▶ 🟩 Ajoneuvo saapuu liikenneympyrään oikeasta alakulmasta | Normal | Metadata on channel Pitäjänmäen liikenneympyrä |
| ▼ 🟩 Car leaving from the roundabout to the LEFT | Normal | Metadata on channel Pitäjänmäen liikenneympyrä |

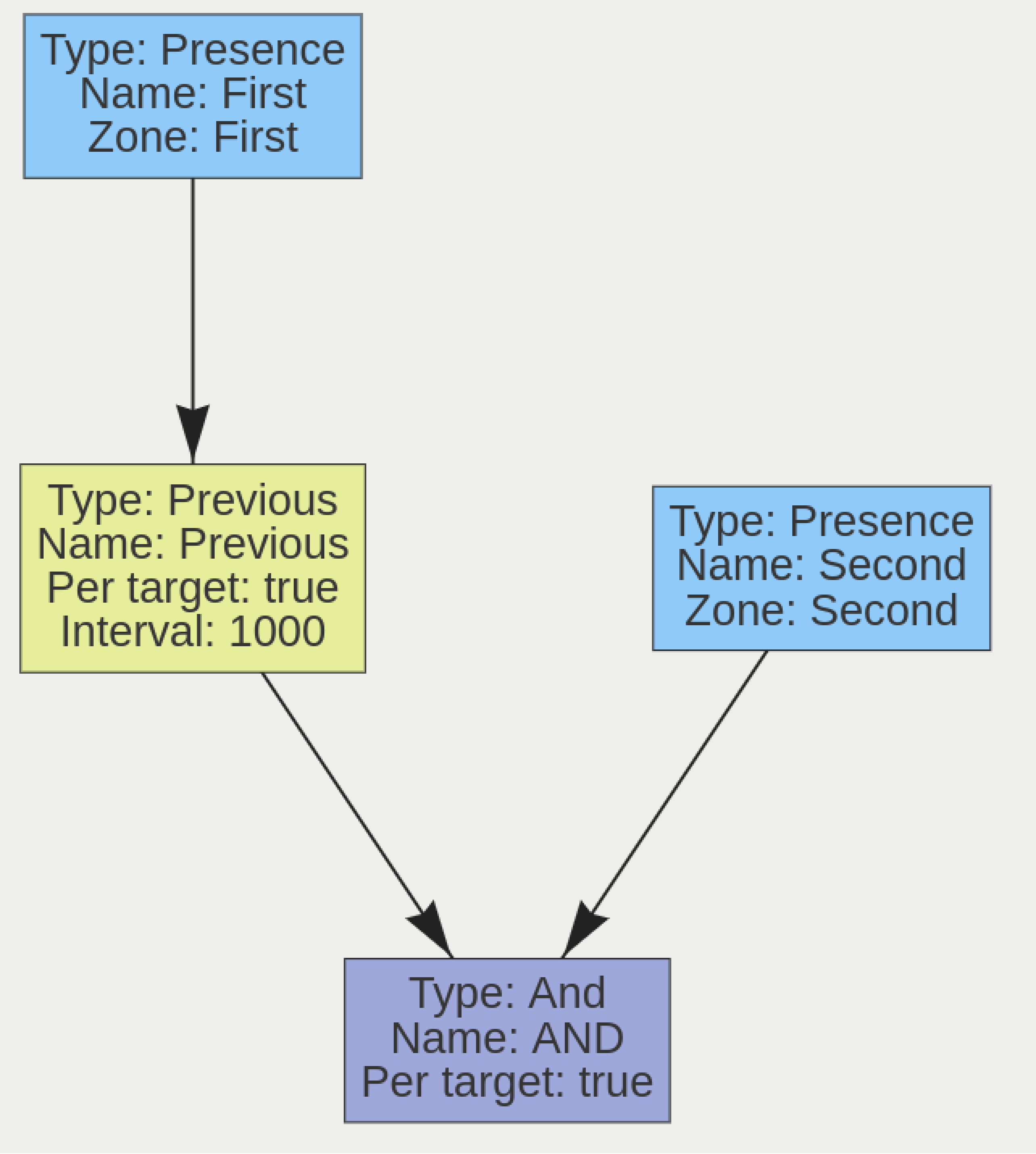
Name: Car leaving from the roundabout to the LEFT
Description:
Priority: Normal
Requires Acknowledgment: No
Viewable in Profiles: Demo
Trigger: Metadata on channel Pitäjänmäen liikenneympyrä
Activate on metadata event Car leaving from roundabout to the LEFT: LEFT, zone 3
Actions: Record video from Pitäjänmäen liikenneympyrä
Resolution: 1920x1080
Recording rate: 15/s
Pre-event recording: Off
Post-event recording: On
Pre-event recording time: 0 s
Post-event recording time: 10 s
Calendar: The alarm is always enabled
Special Days:

## 6.18 Combined Rule Examples

- Double-knock Rule
- Presence in A or B

### 6.18.1 Double-knock Rule

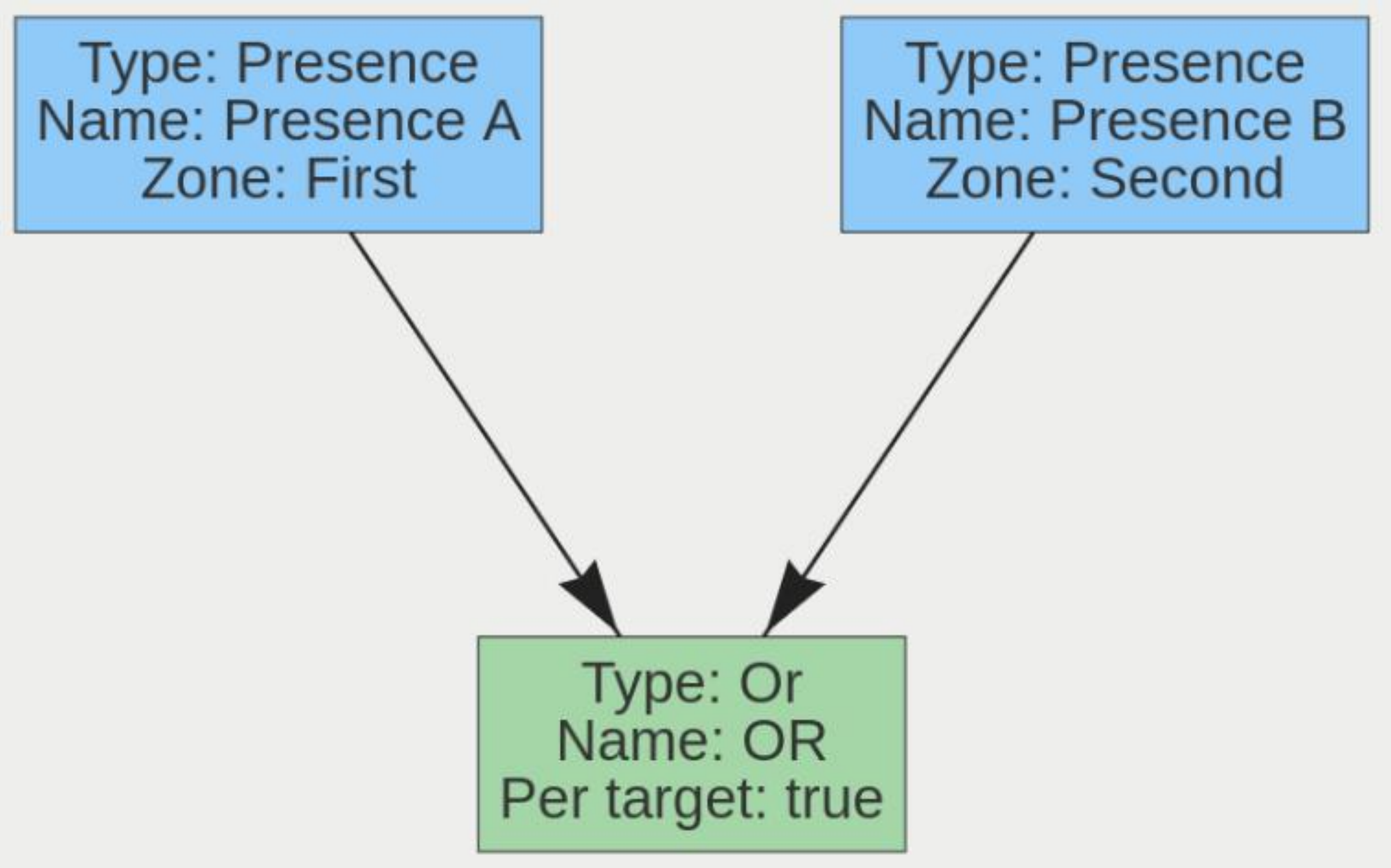
The “double-knock” logical rule triggers when an object enters a zone which had previously entered another defined, zone within a set period of time. The interval on the Previous rule decides how much time can elapse between the object entering the first and second zone. The graph for a double-knock logical rule is as follows:



The rule may be interpreted as follows: “An object is in Zone 2, and was previously in Zone 1 in the last 1000 milliseconds”. This rule can be used as a robust way to detect entry into an area. Since the object has to enter two zones in a specific order, it has the ability to eliminate false positives that may arise from a simple Presence rule.

### 6.18.2 Presence in A or B

This rule triggers when an object is present in either Zone A or Zone B. Its graph is as follows:



A typical use case for this rule is having multiple areas where access is prohibited, but the areas cannot be easily covered by a single zone. Two zones can be created, associated with two separate Presence rules, and they can then be combined using an Or rule.

## 6.19 GPU Performance

Here is a basic explanation of what the CPU is used for. The CPU is used for the following

- Decoding the incoming RTSP stream

- Encoding any outgoing annotated RTSP
- Resizing frames before being passed to the analytic engine
- Preprocessing before the frame is passed to the GPU for the DL tracker to process

The first 3 points are currently performed for all trackers, the standard motion object tracker and the DL trackers.

The last point is performed when using the DL trackers and requires some additional resources from the CPU. As a result, the number of channels that can be supported on a particular CPU is reduced when using the DL trackers.

This chart is giving some overview how VCA can perform with GPU. Different scenarios may affect performance.

| GPU | CUDA cores | Tensor cores | Memory | Processor frequency | Memory Bandwidth (GB/sec) | Actual channels DLOT tested |
|---|---|---|---|---|---|---|
| RTX A4000 | 6144 | 192 | 16 GB | 1750 | 448 | 56 |
| GeForce RTX 3070 | 5888 | 180 | 8 GB | 1440-1710 | 19 | 54 |
| GeForce RTX 2080 Ti | 4352 | 368 | 11GB | 1350-1545 | 616 | 50 |
| Tesla T4 | 2560 | 320 | 16GB |  | 320 | 45 |
| GeForce GTX 1660 SUPER | 1408 |  | 6 GB | 1530-1785 | 336 | 28 |
| GeForce GTX 1650 | 896 |  | 4 GB | 1485-1665 | 128 | 18 |