MIRASYS
DEEP VISION DATA COMPANY

Mirasys VCA Guide

Table of Contents

1. Mirasys VCA Guide

Mirasys VCA (Video Content Analytics) comprises a set of real-time video analytics solutions that utilises 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:

1. **Motion object tracking:** Motion-data based object highlighting and tracking, auto-zoom functionality.
    a. The motion data is produced by server-based, hermeneutic motion detection.
2. **Tripwire counting:** In addition to motion object tracking functionality, line counting for over-head installed cameras, and Spotter client-based counter visualisation.
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.
    a. Multiple filters and rules are supported on any combination of multiple overlapping detection zones, in addition to an advanced people tracking engine optimised for tracking people in cluttered indoor scenes such as retail scenarios. Includes specific high accuracy counting functions optimised for use in busy scenes.
4. **Related analytics options:** Available as separate applications, products or through project-based integrations:
    a. Camera-based (built-in, edge) analytics support selected camera manufacturers and their functionality through manufacturer-specific integration connectors.
    b. Audio analytics technologies refer to software for extracting information and meaning from audio signals, such as detecting sounds of breaking glass, etc.
    c. 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.

d. Number plate recognition technologies (ANPR/LPR) refer to software or camera features for automatically identifying vehicle or container numbers.

Next

2. Getting started

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

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.

Previous Next

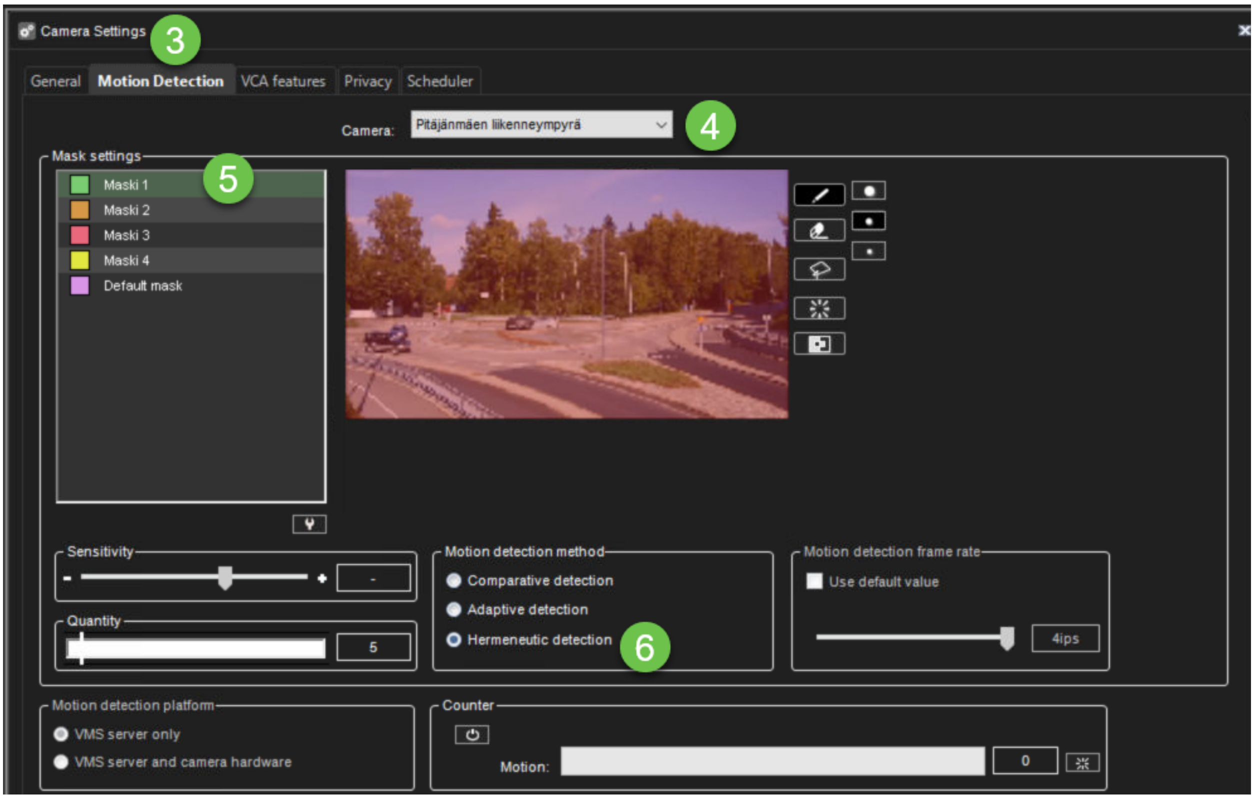
3. Enabling Mirasys VCA

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

Previous Next

3.1. Setting up the Motion detection method

1. Go to the **VMS servers tab**
2. Open **Cameras**
3. Open **Motion Detection**
4. Select camera
5. Select mask 1
6. Select **Hermeneutic detection**



Camera: Pitäjänmäen liikenneympyrä

Mask settings

- 🟩 Maski 1
- 🟧 Maski 2
- 🟥 Maski 3
- 🟨 Maski 4
- 🟪 Default mask

Sensitivity: -

Quantity: 5

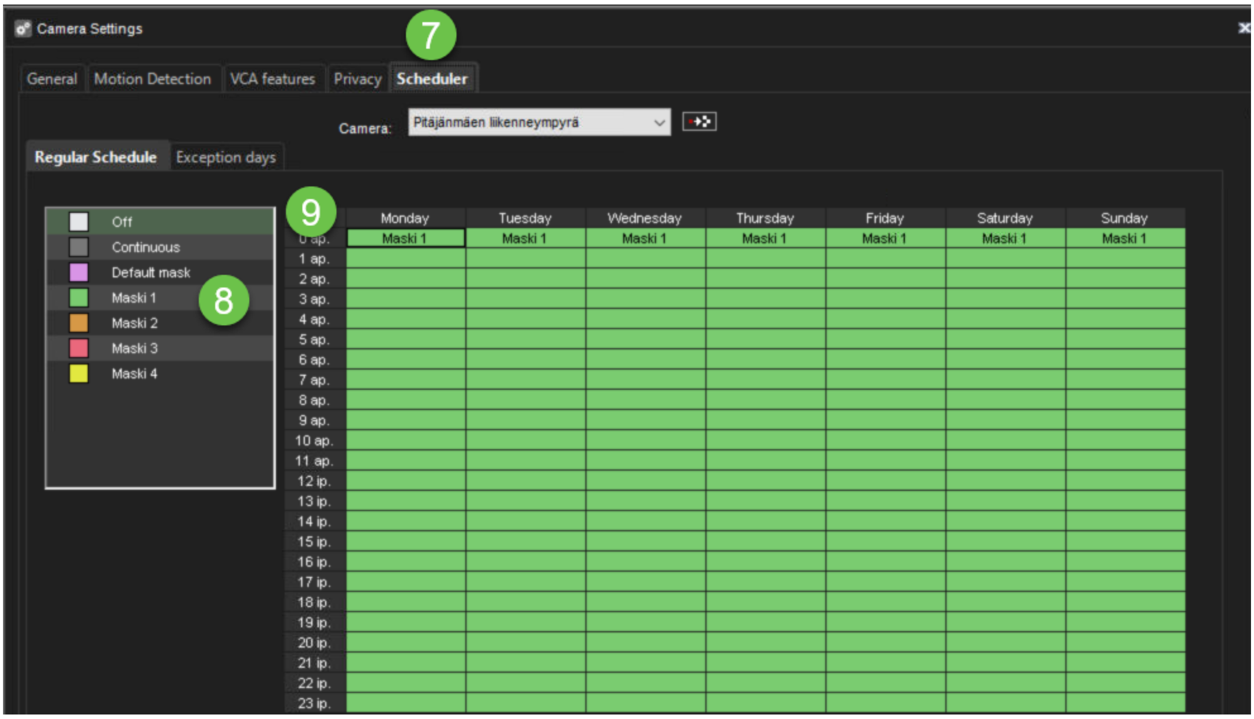
Motion detection method
○ Comparative detection
○ Adaptive detection
● Hermeneutic detection

Motion detection frame rate
□ Use default value
4ips

Motion detection platform
○ VMS server only
○ VMS server and camera hardware

Counter
Motion: 0

7. Click **Scheduler**
8. Select **mask 1**
9. Enable **mask 1** to 24/7
10. Click Ok



Camera: Pitäjänmäen liikenneympyrä

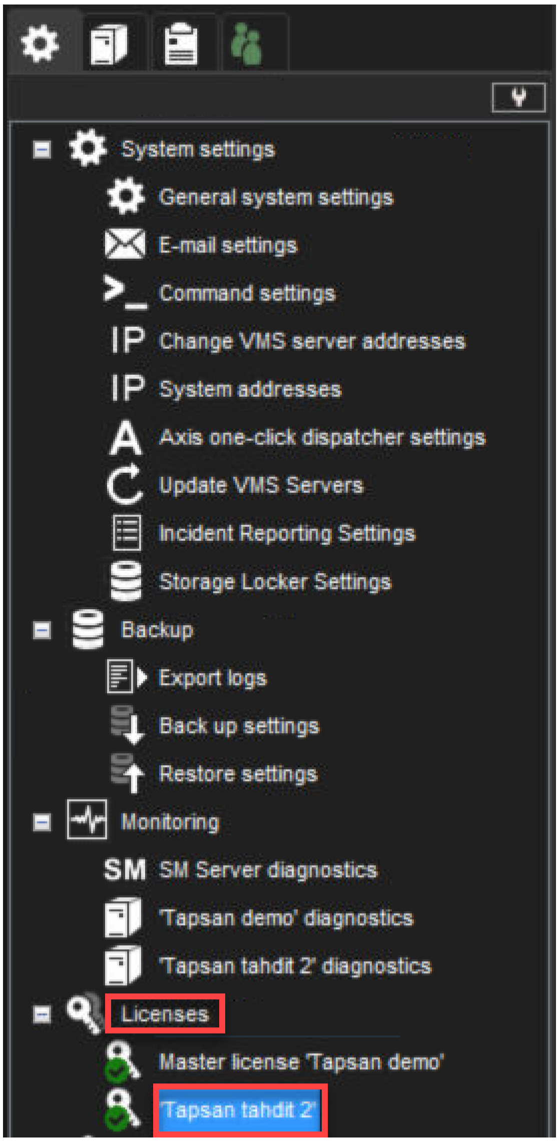
⬜ Off
⬛ Continuous
🟪 Default mask
🟩 Maski 1
🟧 Maski 2
🟥 Maski 3
🟨 Maski 4

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0 ap.	Maski 1	Maski 1	Maski 1	Maski 1	Maski 1	Maski 1	Maski 1
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.							

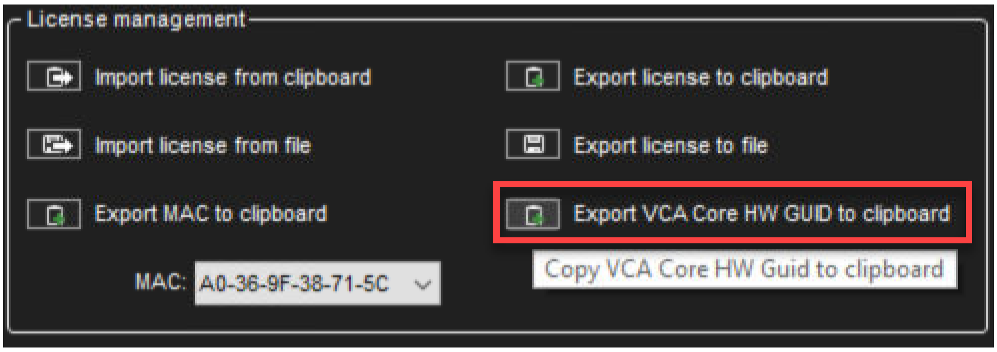
Top Previous Next

3.2. Exporting VCA Core HW GUID file

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

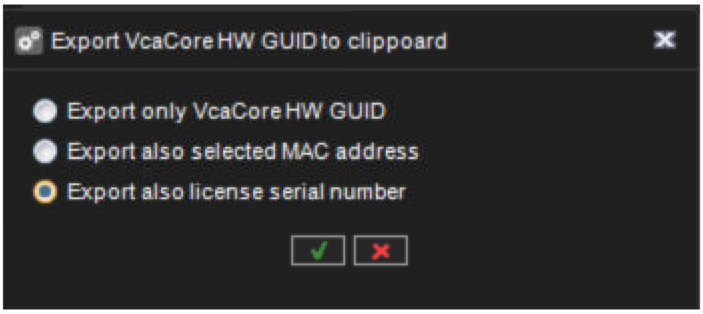


1. Select Export VCA Core HW GUID to clipboard


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
Copy VCA Core HW Guid to clipboard

MAC: A0-36-9F-38-71-5C

1. Select **Export also license serial number**
2. Click Ok



○ Export only VcaCore HW GUID
○ Export also selected MAC address
● Export also license serial number

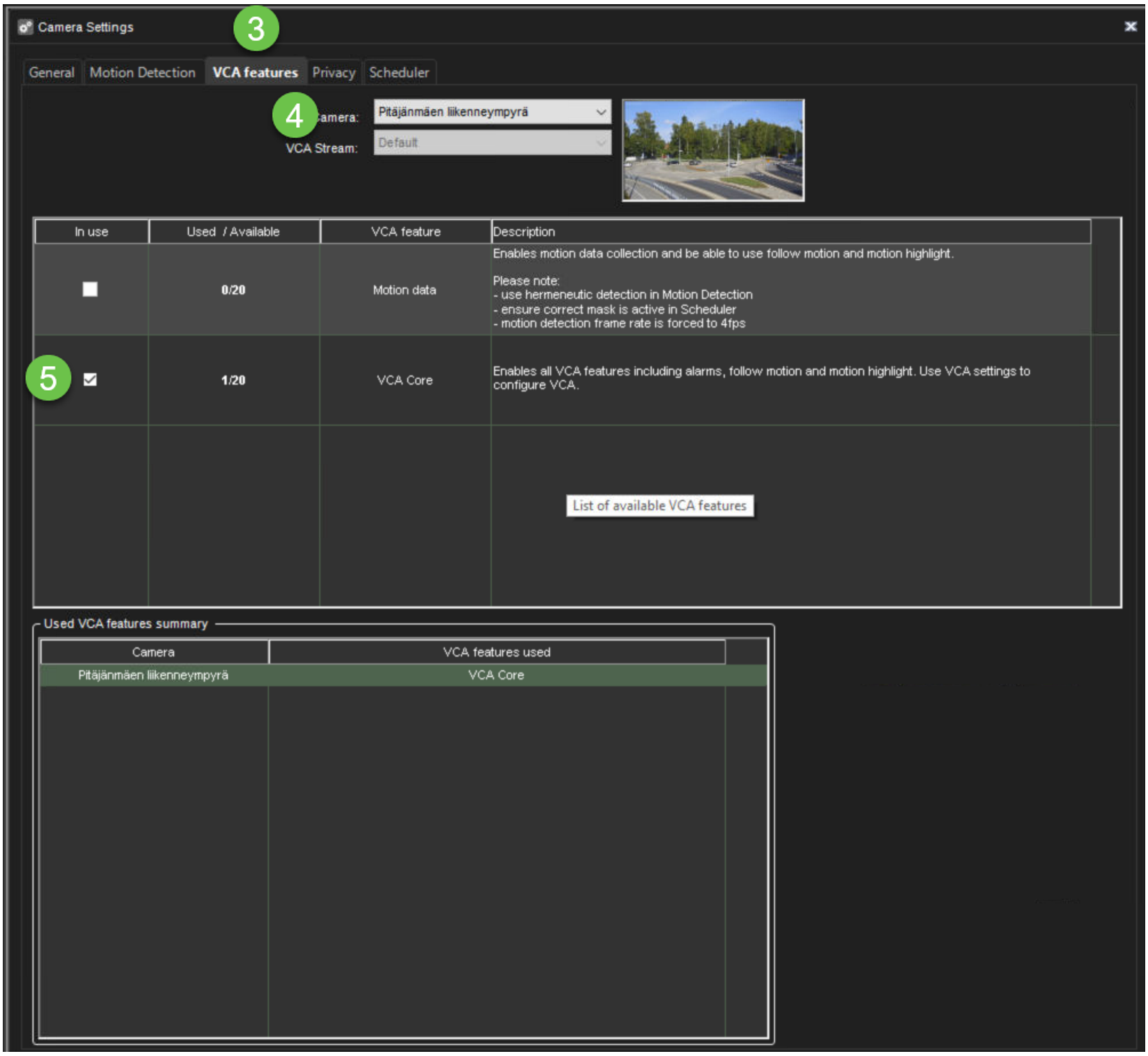
1. Paste clipboard data to the text document.
2. Send it to Mirasys (orders@mirasys.com) to receive the VCA license.

VcaCore Hw Guid:
C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D
License serial number: YL9QMELM9QK5

Top Previous Next

3.3. 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 Settings

Camera: Pitäjänmäen liikenneympyrä

VCA Stream: Default

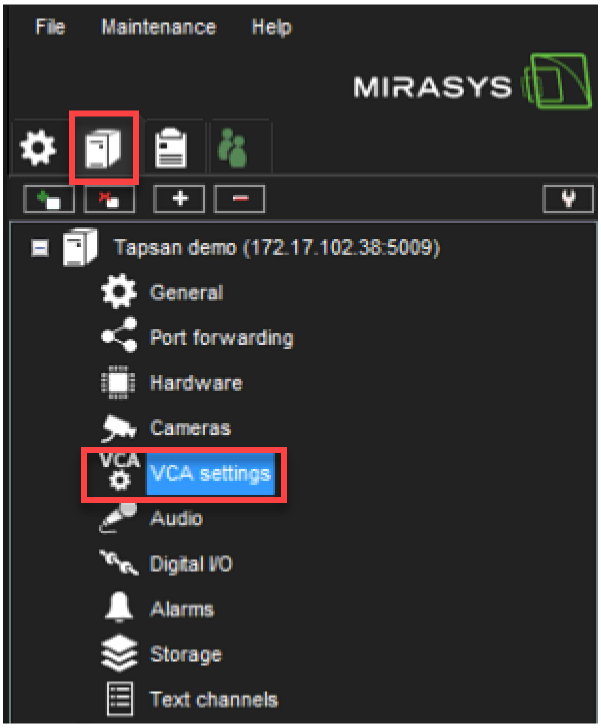
In use	Used / Available	VCA feature	Description
□	0/20	Motion data	Enables motion data collection and be able to use follow motion and motion highlight. Please note: - use hermeneutic detection in Motion Detection - ensure correct mask is active in Scheduler - motion detection frame rate is forced to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

Used VCA features summary

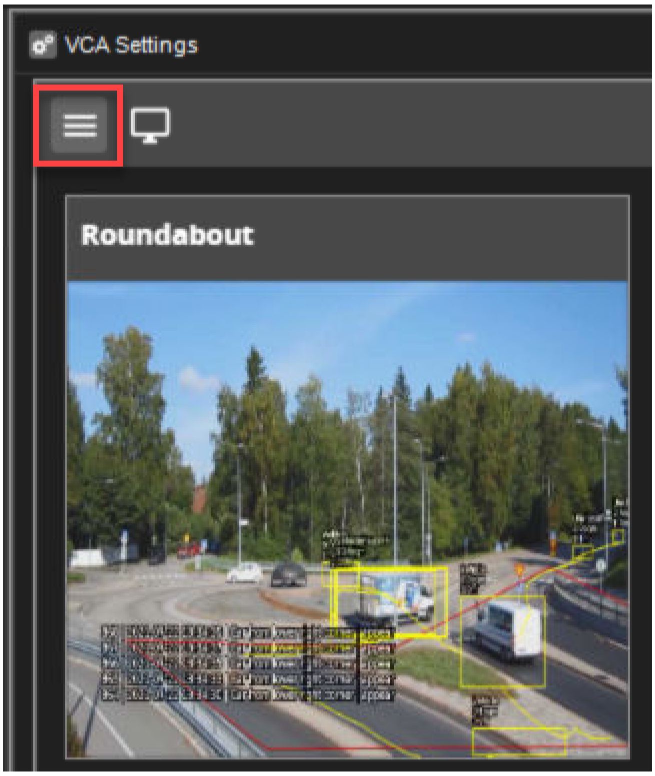
Camera	VCA features used
Pitäjänmäen liikenneympyrä	VCA Core

3.4. Activating VCA licenses

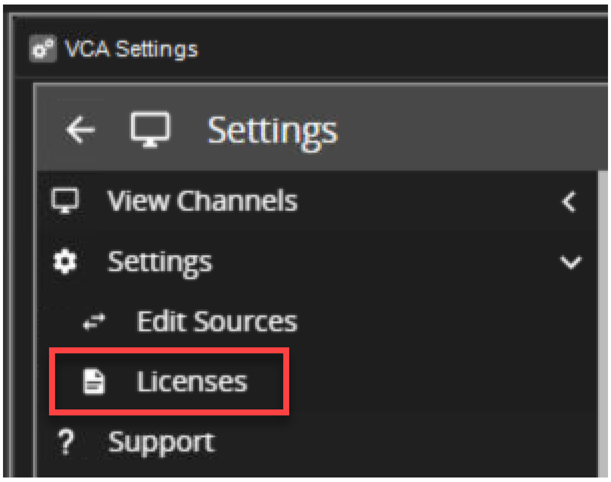
1. Go to the VMS Servers tab
2. Open VCA settings



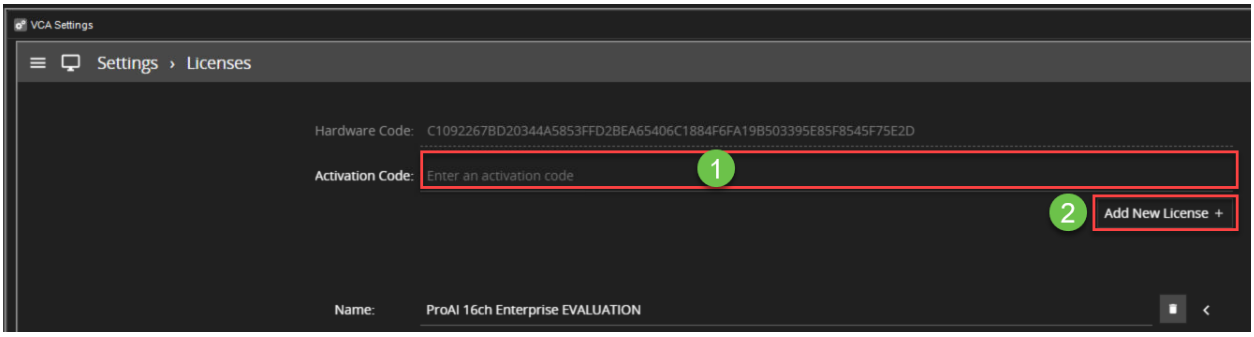
3. Click the **Settings** icon from the upper left corner



5. Select **Licenses**



1. Paste the license, which you have received from Mirasys to the **Activation Code field**
2. Click **Add New License**



VCA Settings

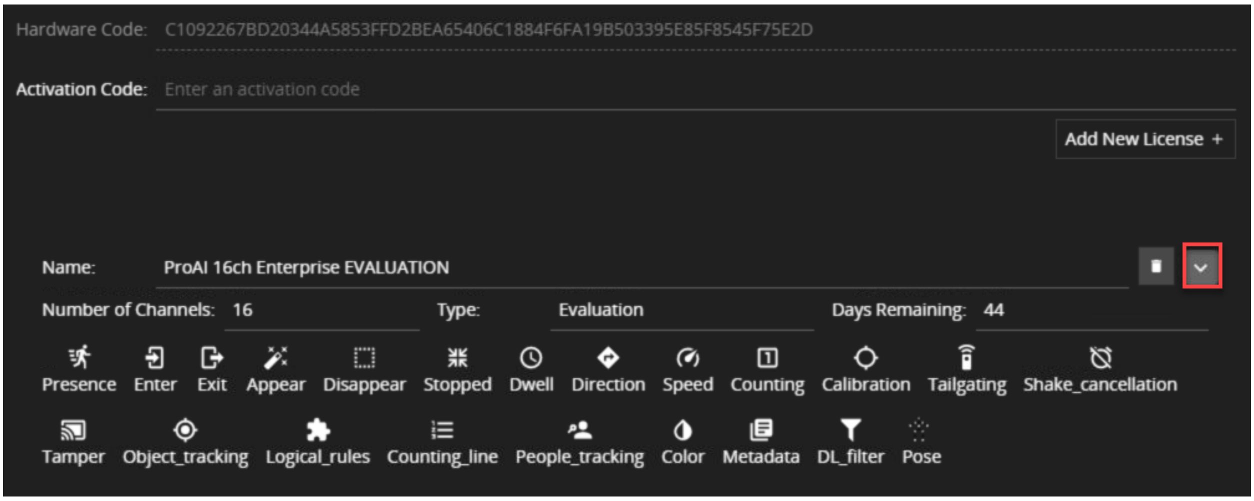
Settings › Licenses

Hardware Code: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

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

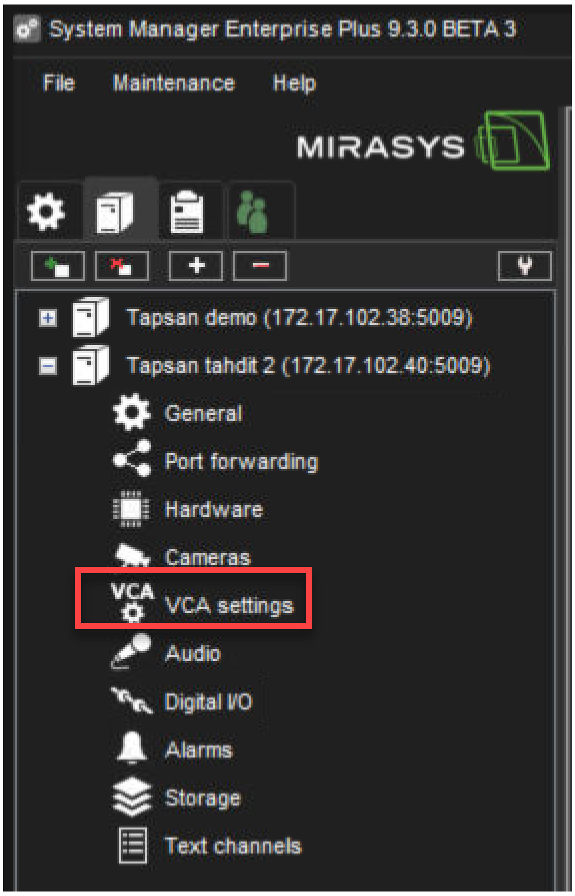
Name: ProAI 16ch Enterprise EVALUATION

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

Top Previous Next

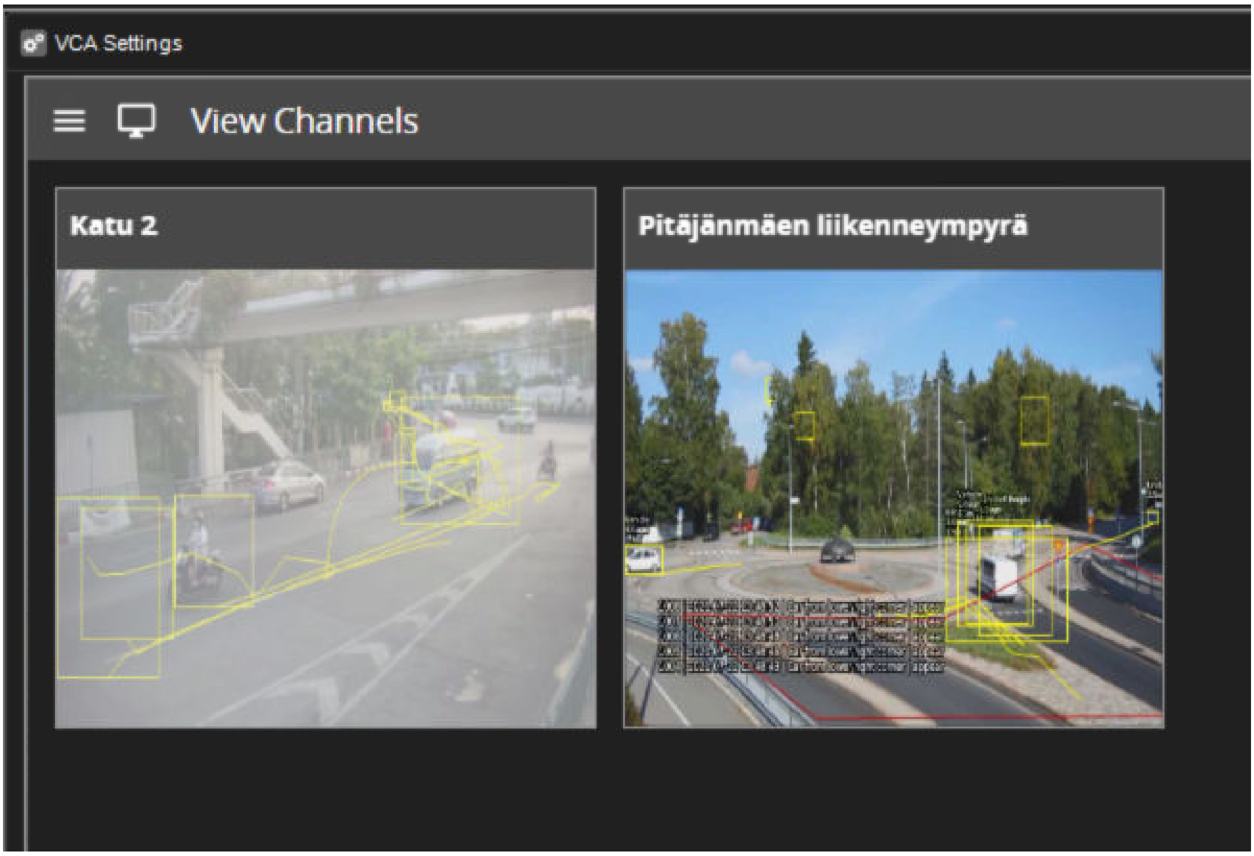
4. VCA settings

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



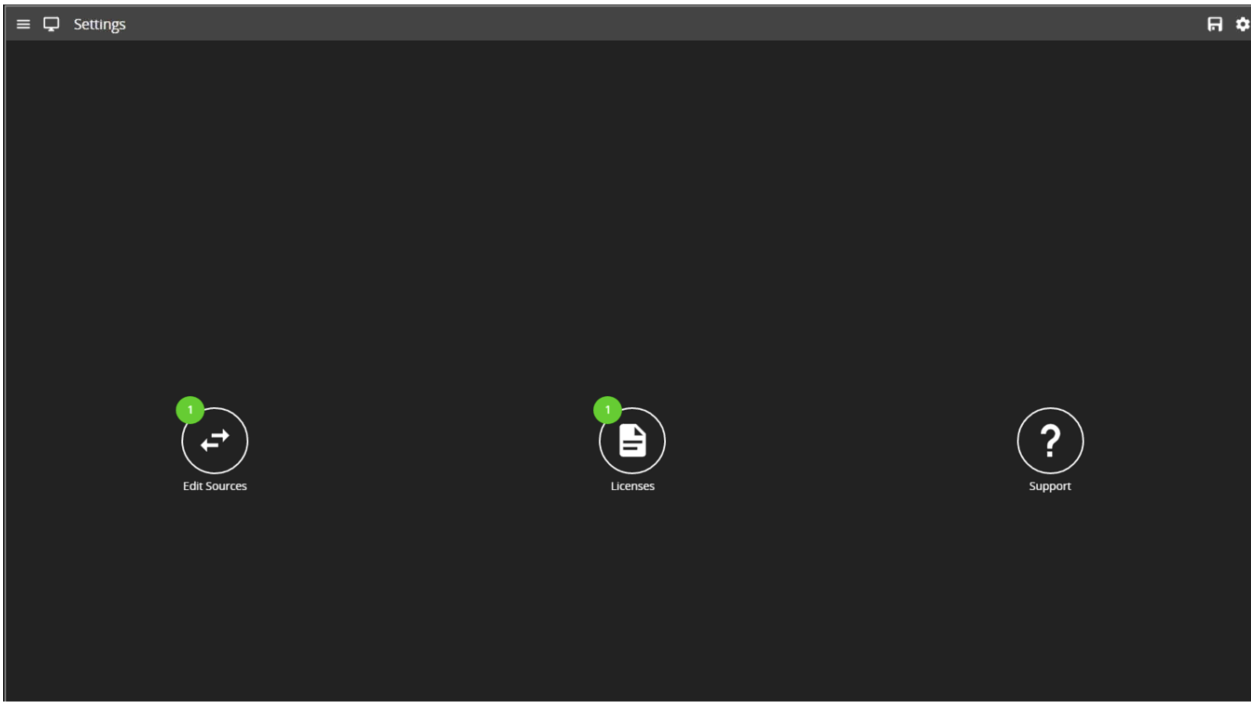
View Channels

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



Settings

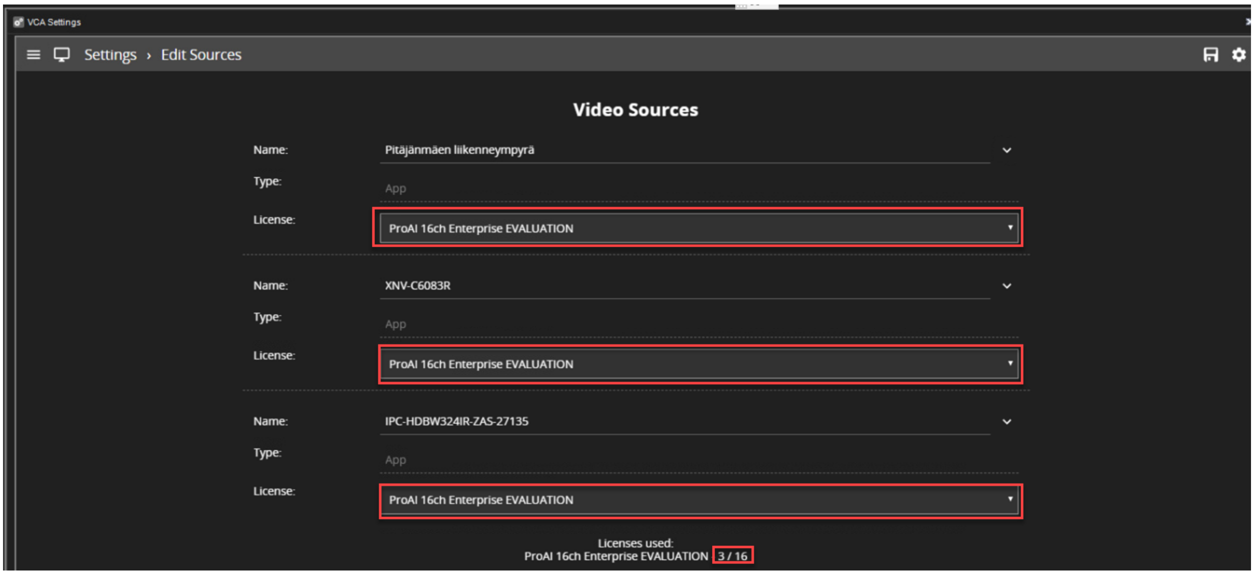
- Edit Sources
- Licenses
- Support



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

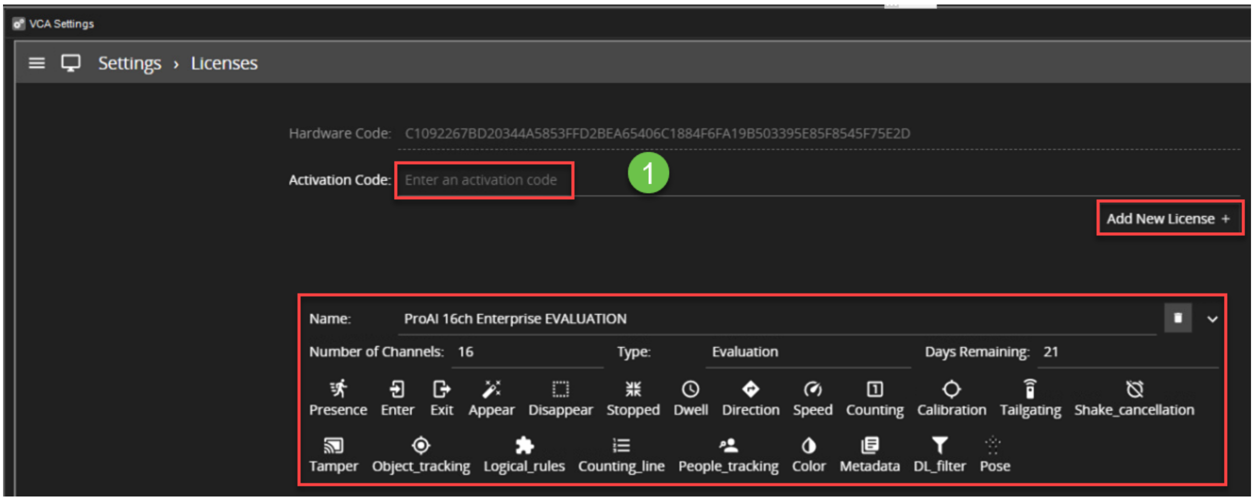
Name:	Pitäjänmäen liikenneympyrä
Type:	App
License:	ProAI 16ch Enterprise EVALUATION
Name:	XNV-C6083R
Type:	App
License:	ProAI 16ch Enterprise EVALUATION
Name:	IPC-HDBW324IR-ZAS-27135
Type:	App
License:	ProAI 16ch Enterprise EVALUATION

Licenses used:
ProAI 16ch Enterprise EVALUATION 3 / 16

Licenses

Licenses show existing license type and that features

Users can add more VCA licenses



Hardware Code: C1092267BD20344A5853FFD2BEA65406C1884F6FA19B503395E85F8545F75E2D

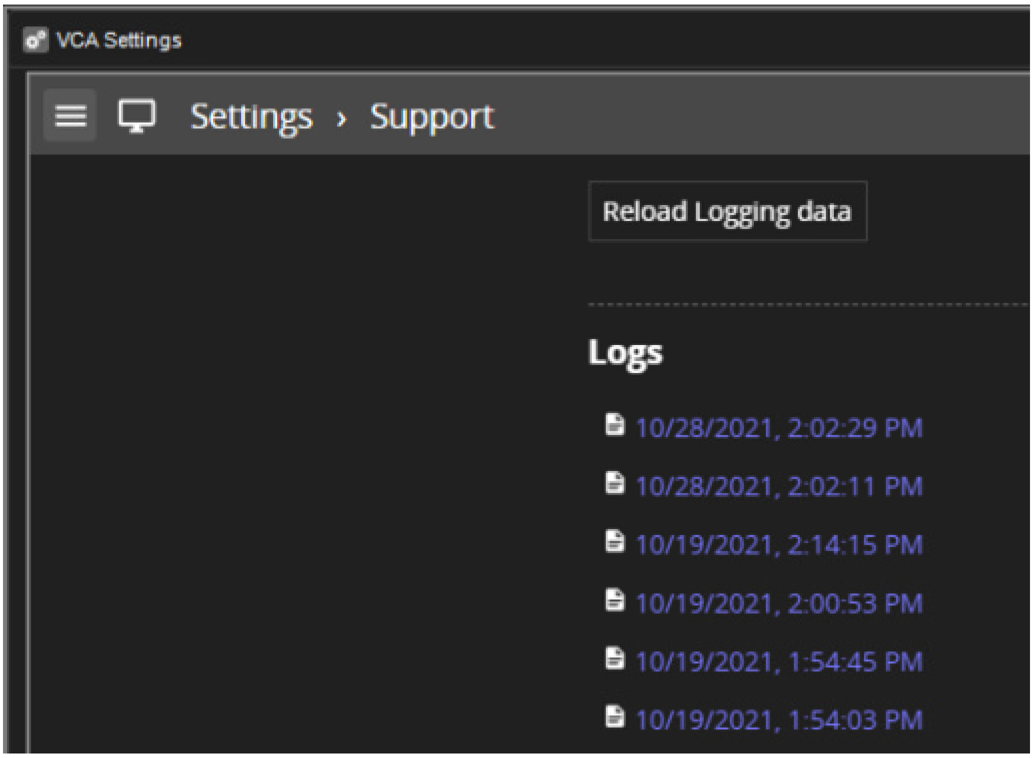
Activation Code: Enter an activation code

Name: ProAI 16ch Enterprise EVALUATION

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

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

Previous Next

5. 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.

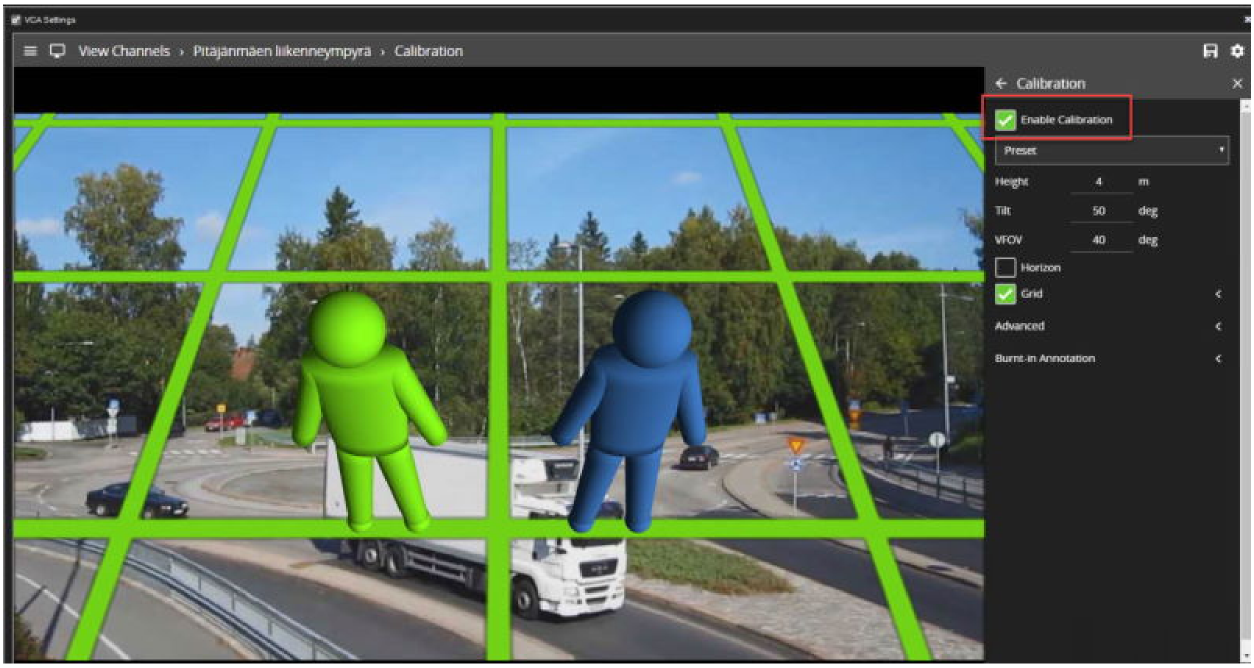
Camera calibration is split into the following sub-topics:

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

Previous Next

5.1. 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

Calibration Controls

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.

Mouse Controls

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

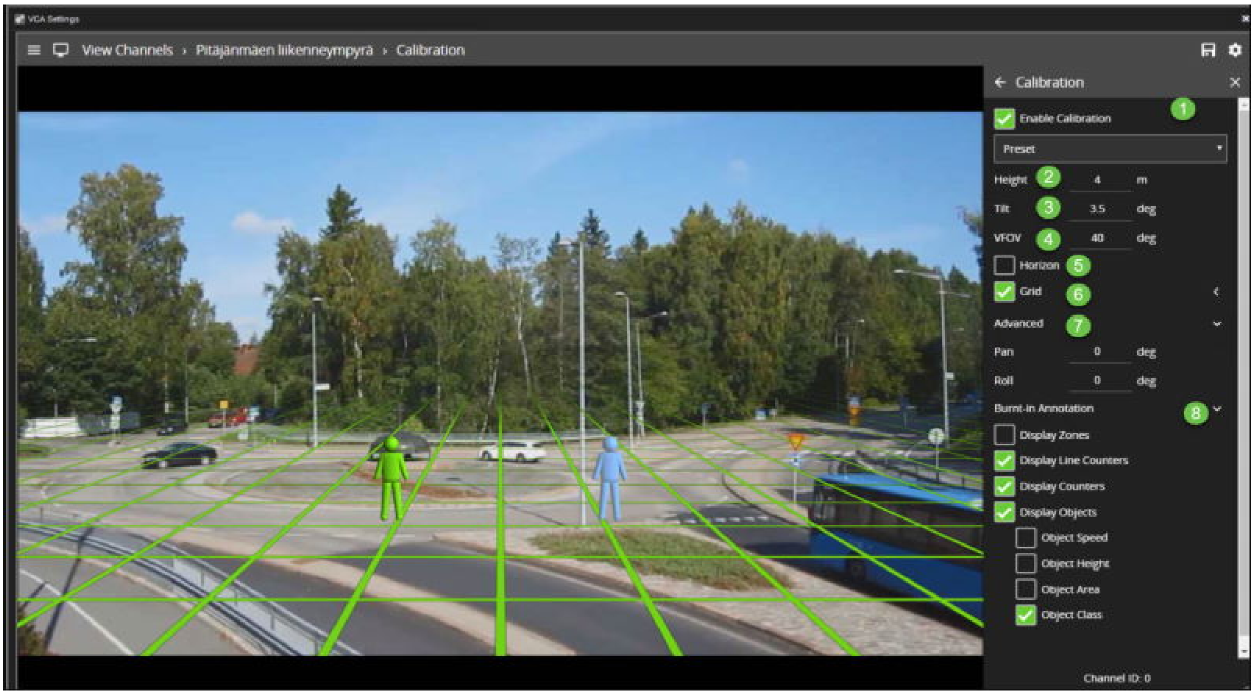
- Click and drag the ground plane to change the camera tilt angle.

- 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.

Control Panel Items

1. The control panel (shown on the right-hand side in the image above) contains the following controls:
2. Height: Adjusts the height of the camera
3. Tilt: Adjusts the tilt angle of the camera
4. 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.
5. Horizon: Enables/disables the horizon display. Useful to line up against a horizon in a deep scene.
6. 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.
7. Advanced: Exposes advanced settings for controlling the pan and roll of the camera.
8. 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

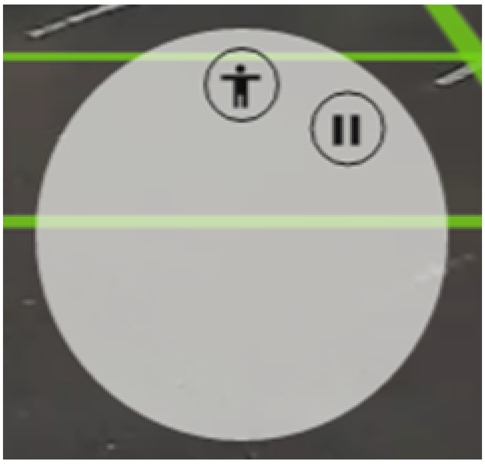
□ Object Area

☑ Object Class

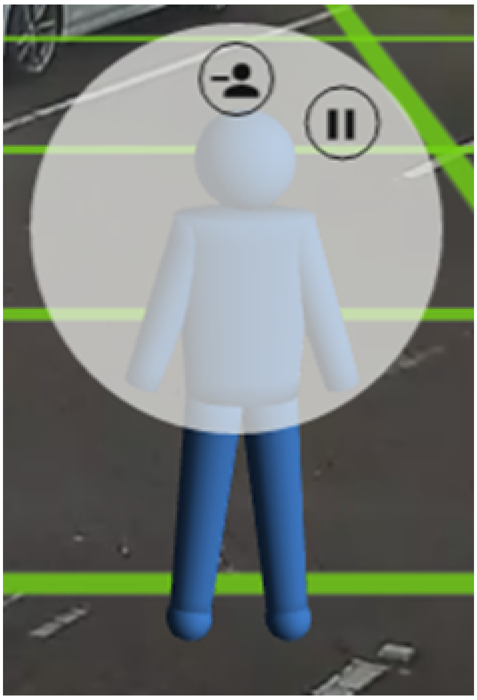
Channel ID: 0

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:

1. ⏸ Pause the video. Pausing the video can make it easier to align mimics up with objects in the scene.
2. ▶ Re-starts playing the video after it was previously paused.
3. 🯅 Adds an extra mimic to the ground plane.

4. Removes the currently selected mimic from the ground plane.

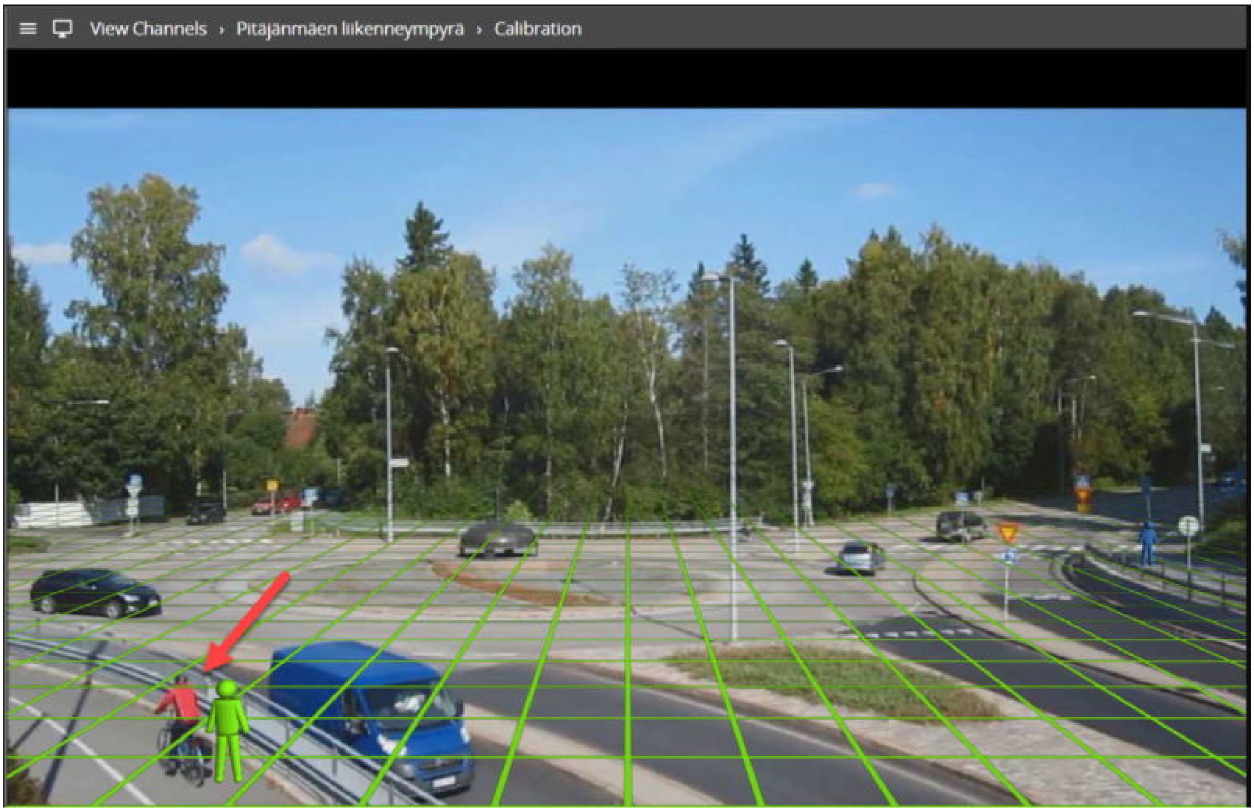
Top Previous Next

5.2. 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.

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:



Step 2: Enter the Camera Vertical Field of View

Mirasys VCA Guide

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.

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.

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.

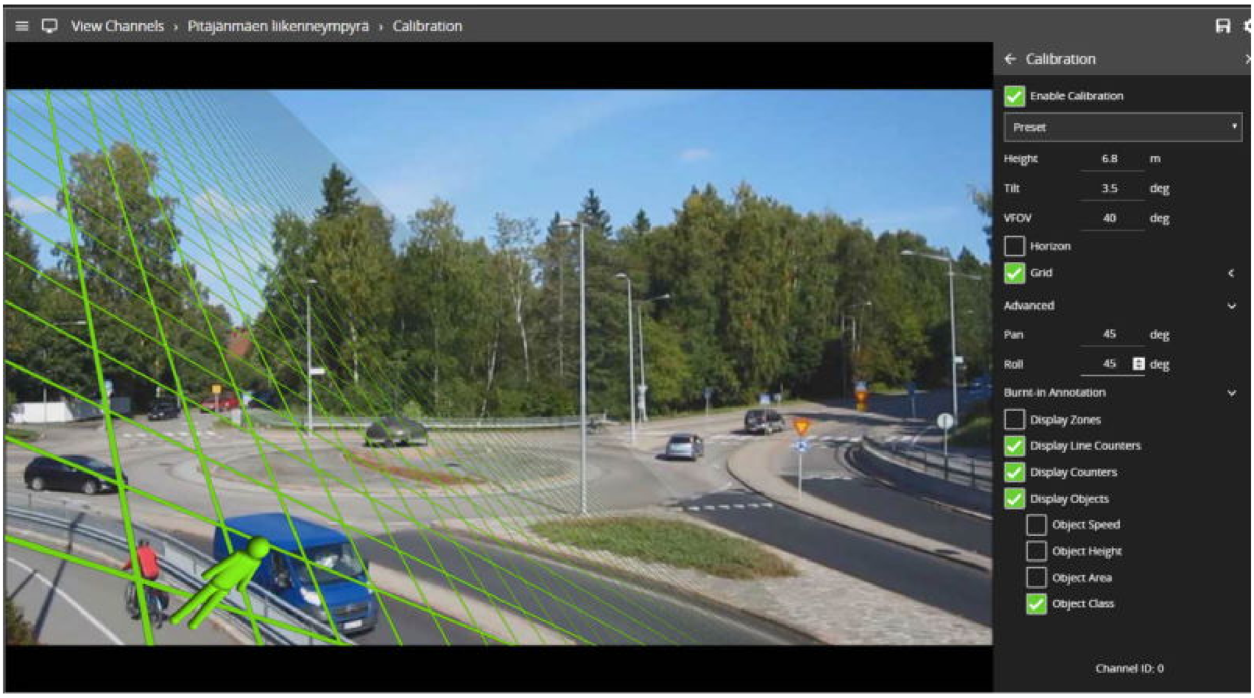
Step 5: Verify the Setup

- 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.

Top Previous Next

5.3. 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 visualise the calibration setup if the scene has a pan or roll with respect to the camera.



View Channels › Pitäjänmäen liikenneympyrä › Calibration

Calibration

☑ 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

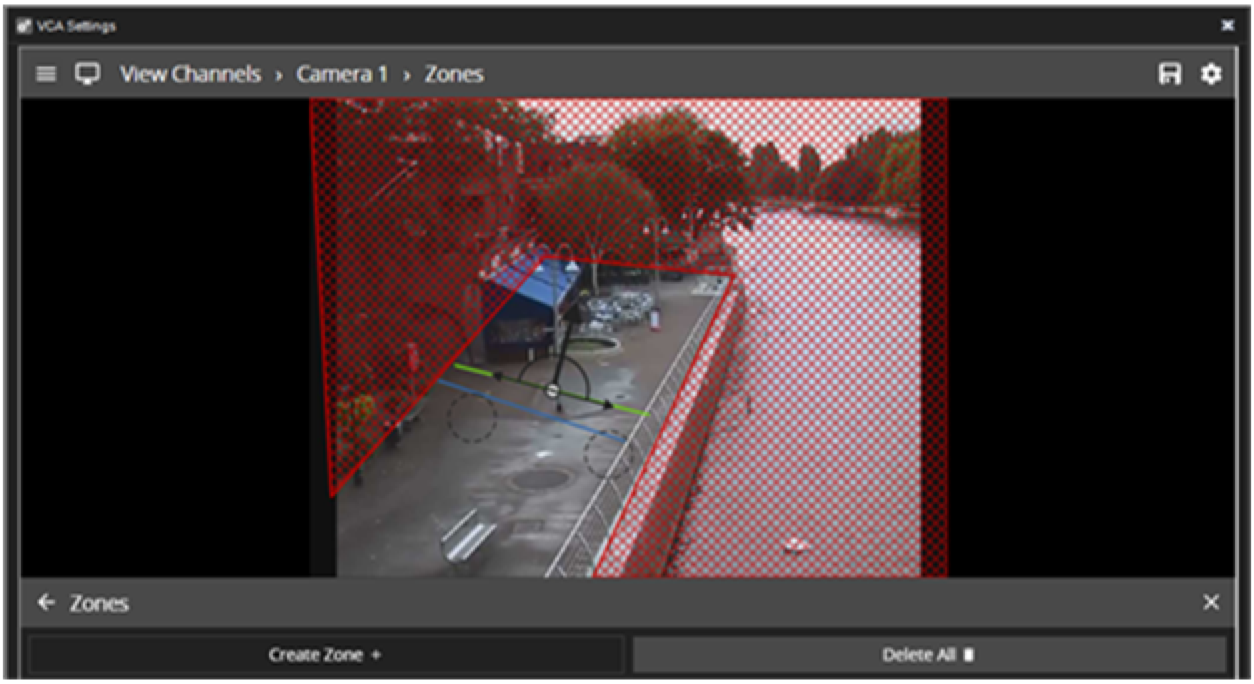
Channel ID: 0

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.

Top Previous Next

6. Zones

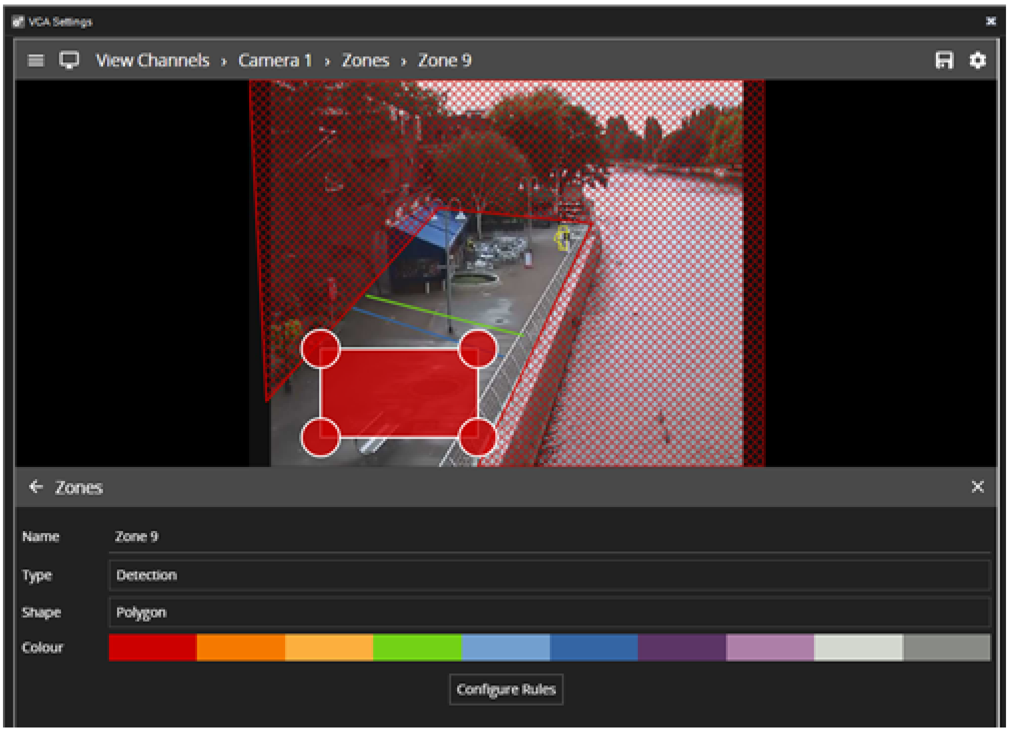
Zones are the detection areas on which VCAcore operate.
To detect a specific behaviour, a zone must be configured to specify the area where a rule applies



Previous Next

6.1. Zone specific settings

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



Name: Zone 9
Type: Detection
Shape: Polygon
Colour

- **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

Top Previous Next

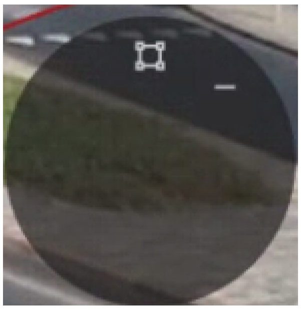
6.2. Adding a zone

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

The context menu

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



The possible actions from the context menu are:

Adds a new zone.

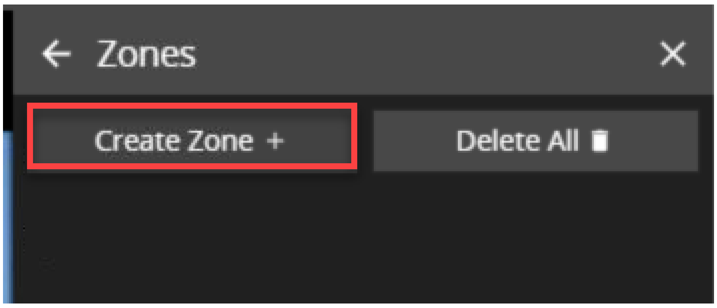
Deletes an existing zone.

Adds a node to a zone.

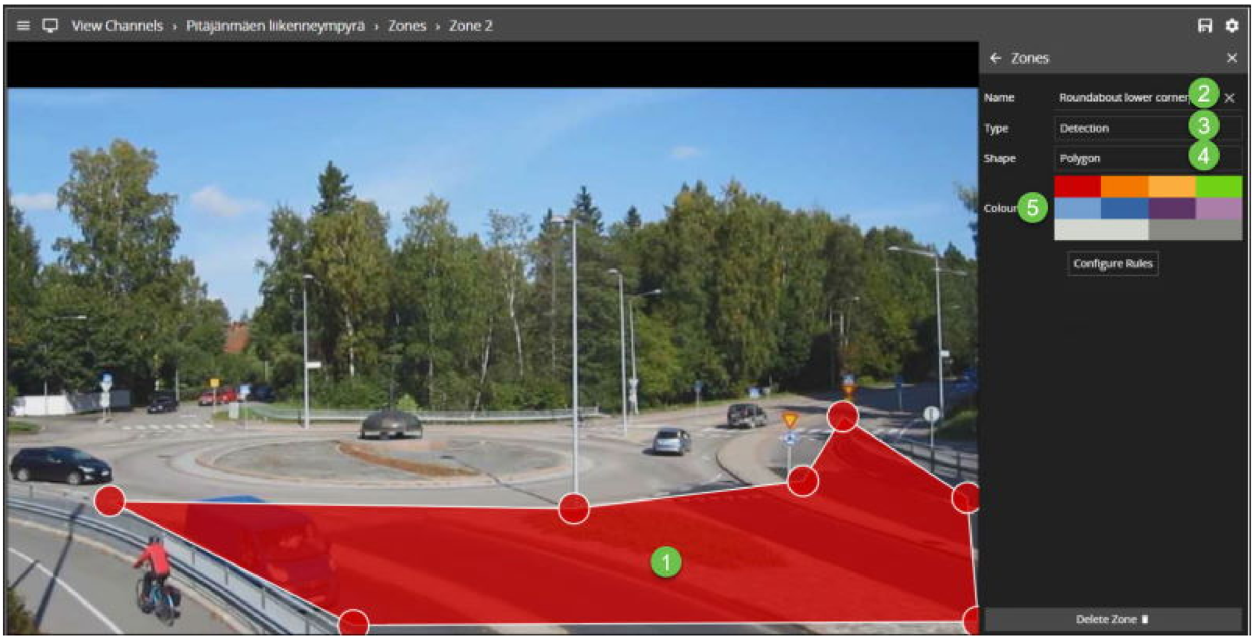
Deletes an existing node from a zone.

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 colour



Name: Roundabout lower corner
Type: Detection
Shape: Polygon
Colour

Top Previous Next

6.3. 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.

Top Previous Next

6.4. 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

Top Previous Next

7. Rules

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

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.

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.

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.
More detail on the differences between these concepts is outlined below:

Previous Next

7.1. Basic inputs

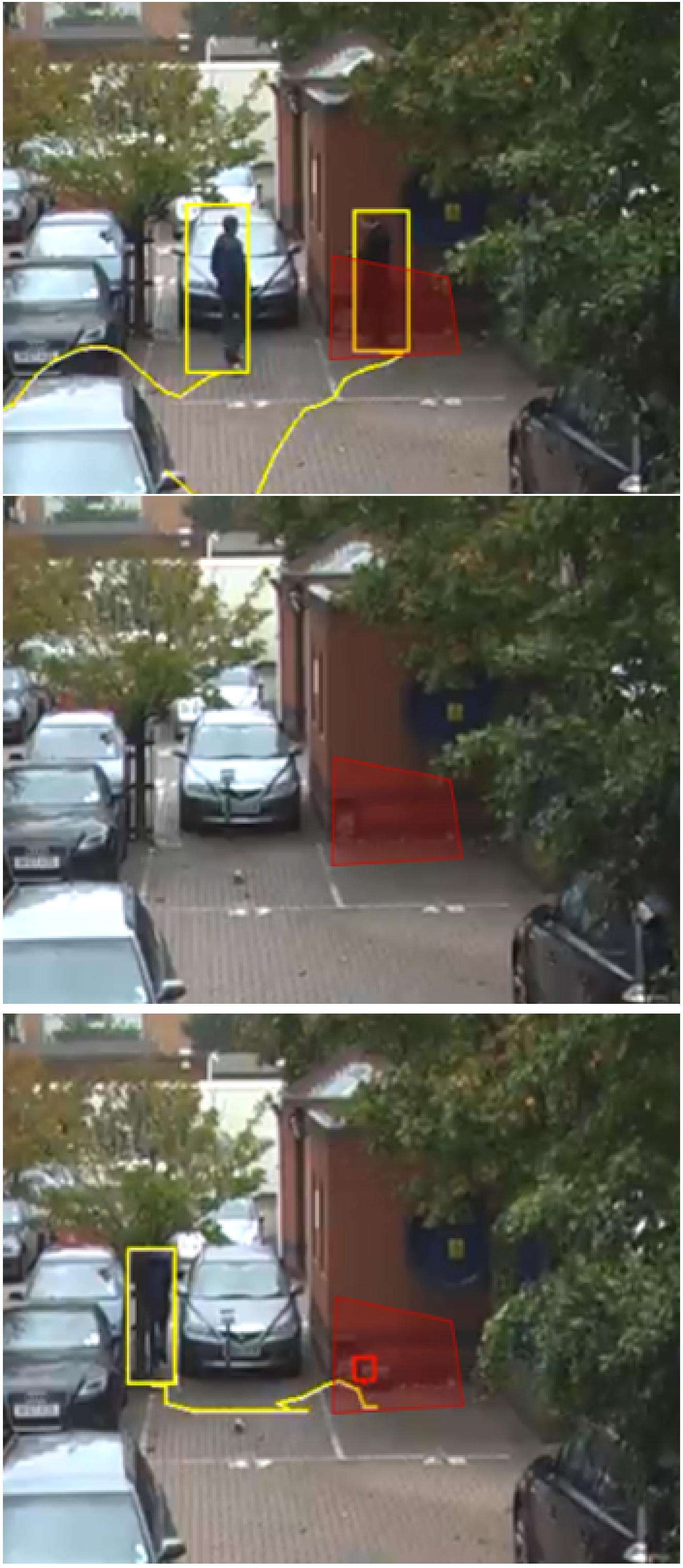
A basic input or rule can only be used to trigger an action or as an input to another rule type.
Basic inputs always require a zone, and potentially some additional parameters.
A basic input can be used on its own to trigger an action, although they are often used as an input to other filters or conditional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

Top Previous Next

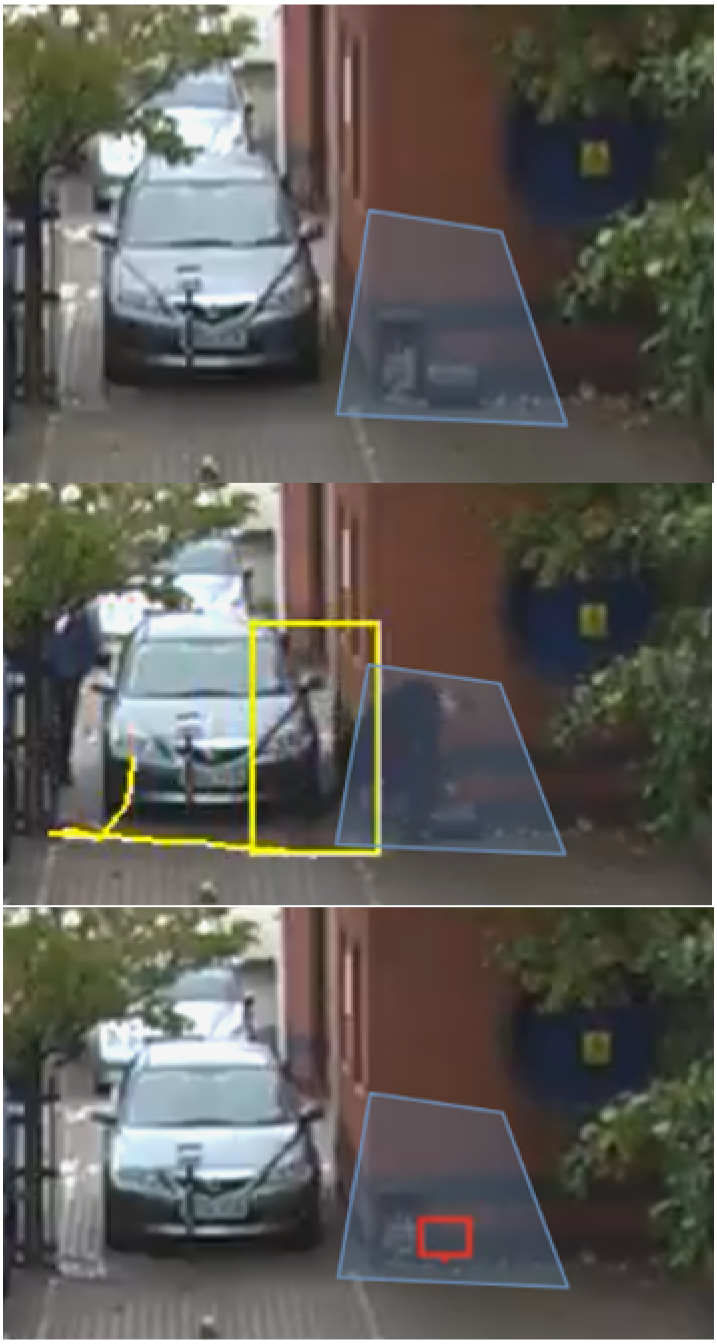
7.2. Abandoned and Removed object

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.

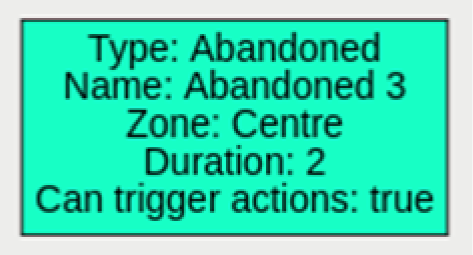


Below is a similar example scenario where the bag is removed from the defined zone resulting in the rule triggering.

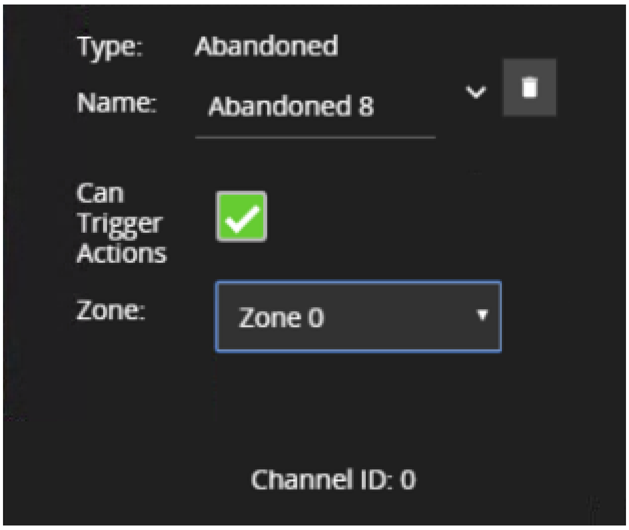


Note: The algorithm used for abandoned and removed object detection is the same in each case, and therefore cannot differentiate between objects which have been abandoned or removed.

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 ☑

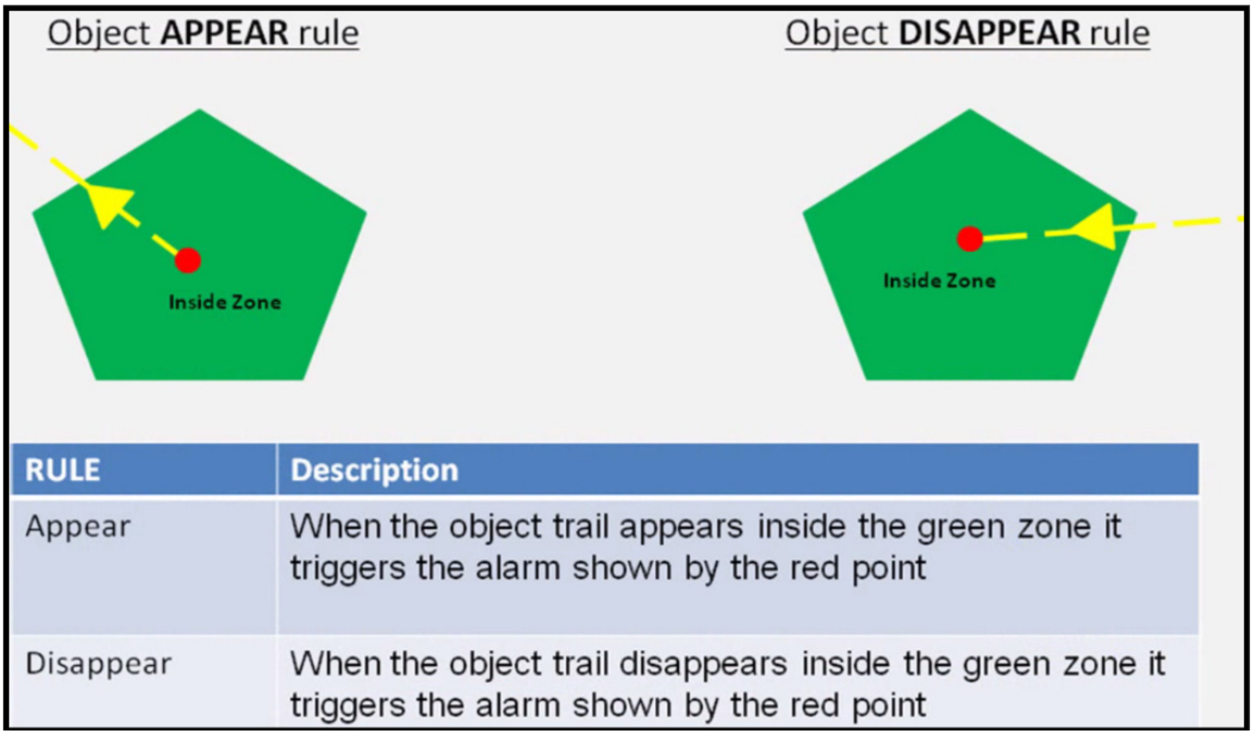
Zone: Zone 0

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

Top Previous Next

7.3. Appear and Disappear

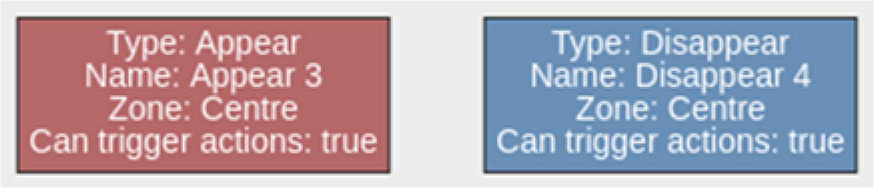


RULE	Description
Appear	When the object trail appears inside the green zone it triggers the alarm shown by the red point
Disappear	When the object trail disappears inside the green zone it triggers the alarm shown by the red point

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

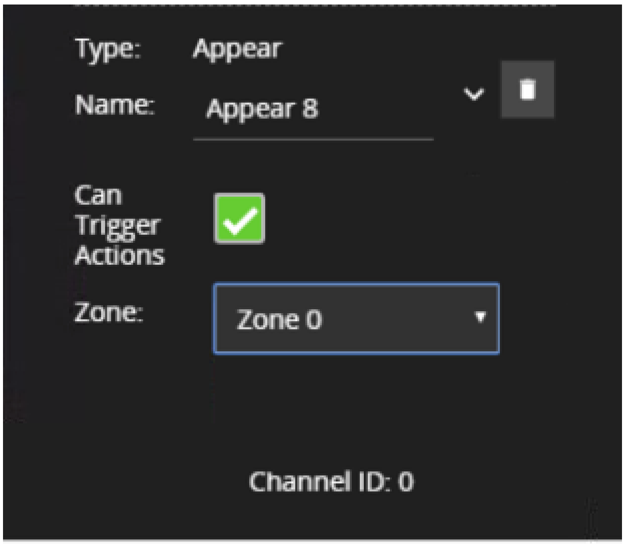
Conversely, the disappear rule detects objects that stop being tracked within a zone, e.g. a person who exits the scene through a doorway.

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



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

Configuration Appear

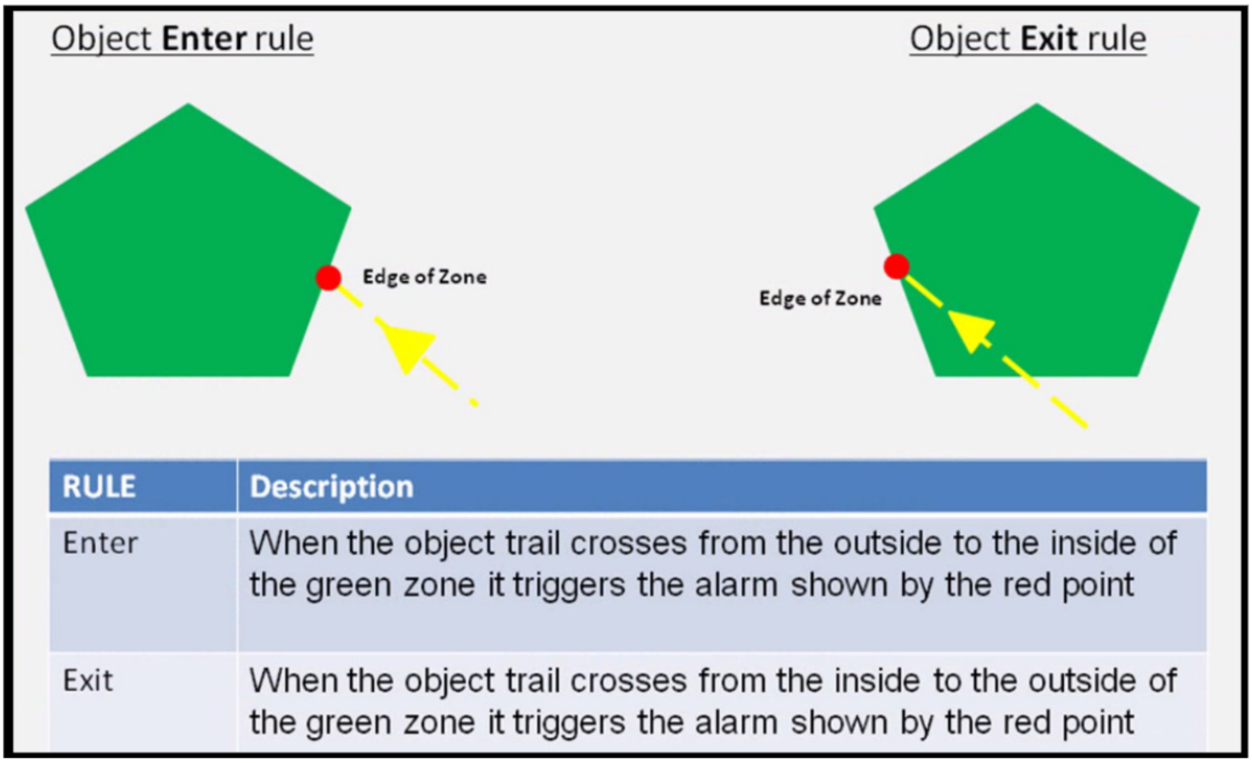
Property	Description	Default Value
Name	A user-specified name for this rule	"Appear #"
Can Trigger Actions	Specifies whether events generated by this rule trigger actions	Active
Zone	The zone this rule is associated with	None

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

Top Previous Next

7.4. Enter and exit

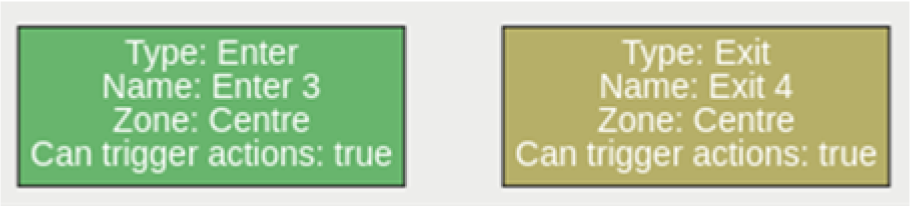


RULE	Description
Enter	When the object trail crosses from the outside to the inside of the green zone it triggers the alarm shown by the red point
Exit	When the object trail crosses from the inside to the outside of the green zone it triggers the alarm shown by the red point

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.

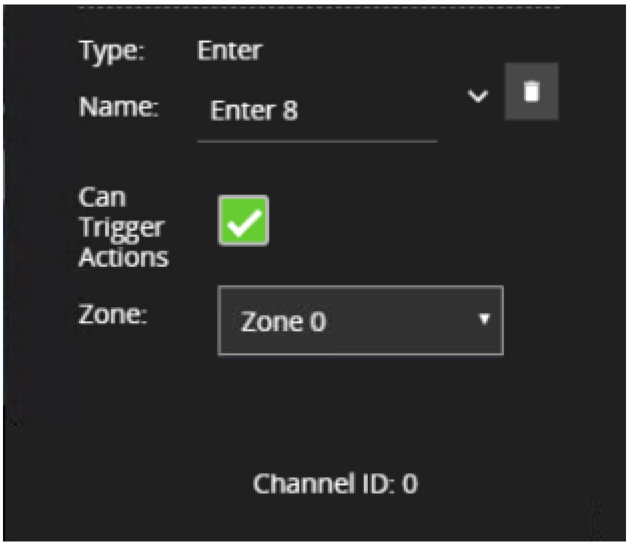
Note: Enter and exit rules differ from appear and disappear rules, as follows:

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



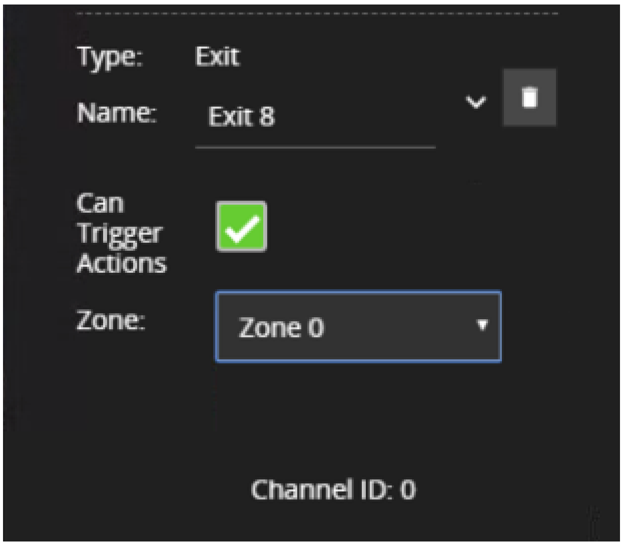
Type: Enter

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

Configuration Enter

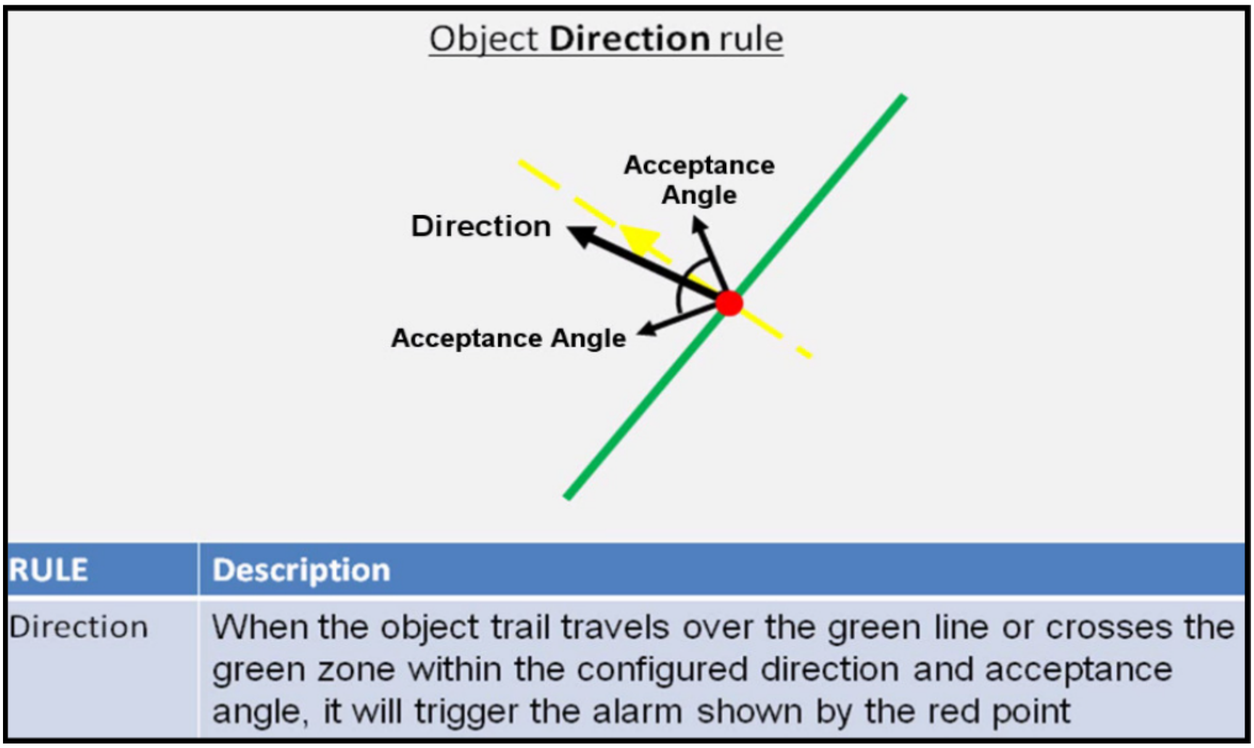
Property	Description	Default Value
Name	A user-specified name for this rule	"Enter #"
Can Trigger Actions	Specifies whether events generated by this rule trigger actions	Active
Zone	The zone this rule is associated with	None

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

Top Previous Next

7.5. 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

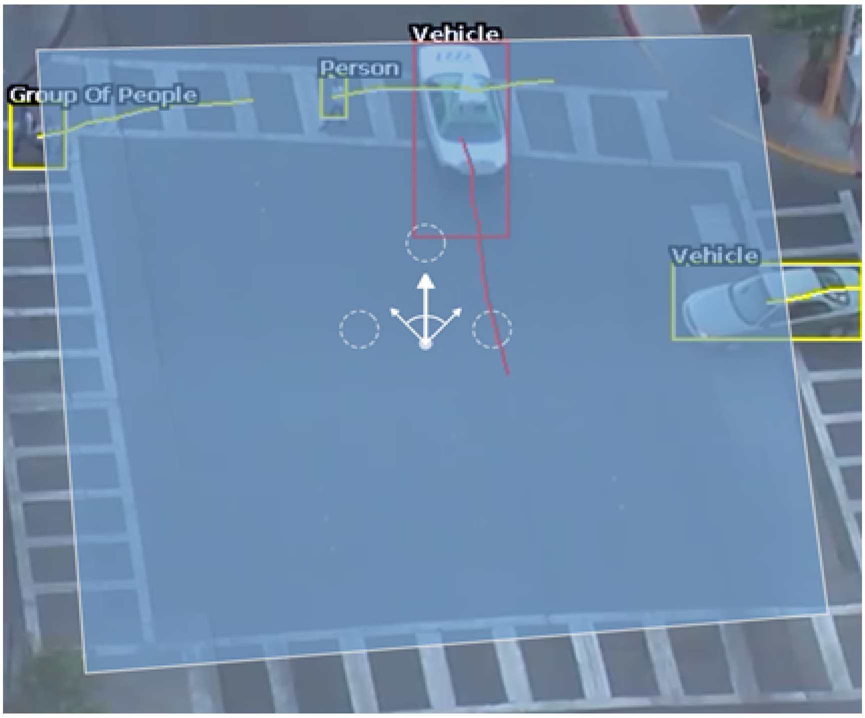
The direction rule detects objects moving in a specific direction.

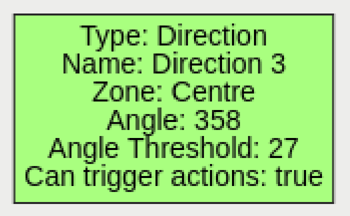
Configure the direction and acceptance angle by moving the arrows on the direction control widget.

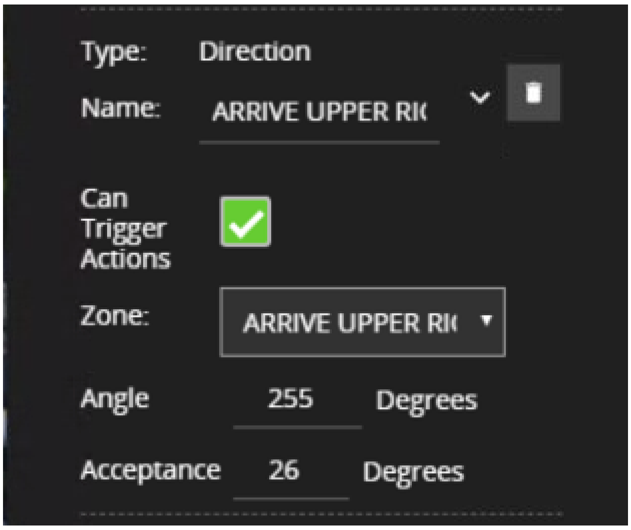
The primary direction is indicated by the large central arrow. The acceptance angle is the angle between the two smaller arrows.

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

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

Top Previous Next

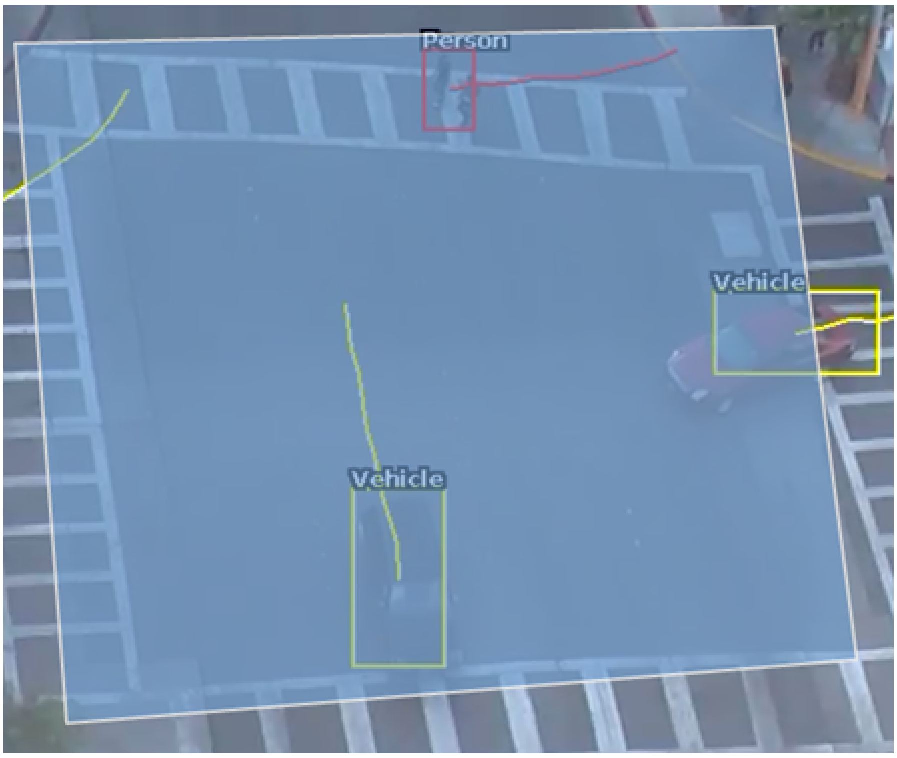
7.6. Dwell

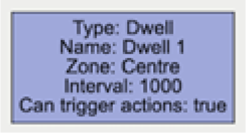
A dwell rule triggers when an object has remained in a zone for a specified amount of time.

The interval parameter is the time the object has to remain in the zone before an event is triggered.

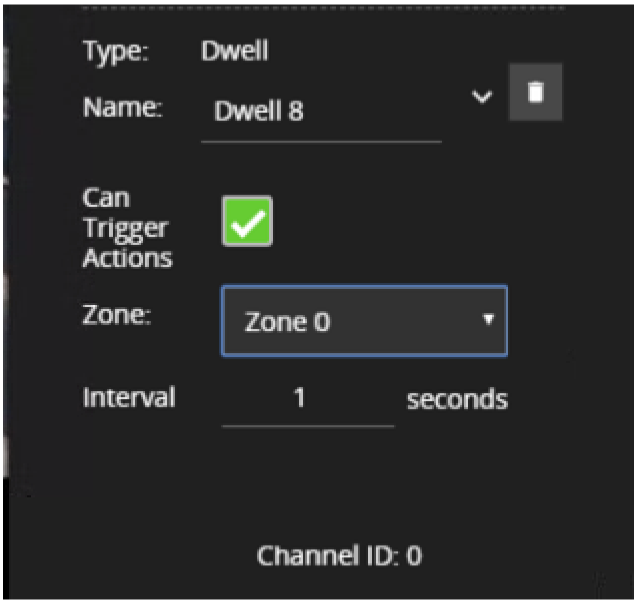
The following image illustrates how the person detected in the zone is highlighted red as they have dwelt in the zone for the desired period of time.

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

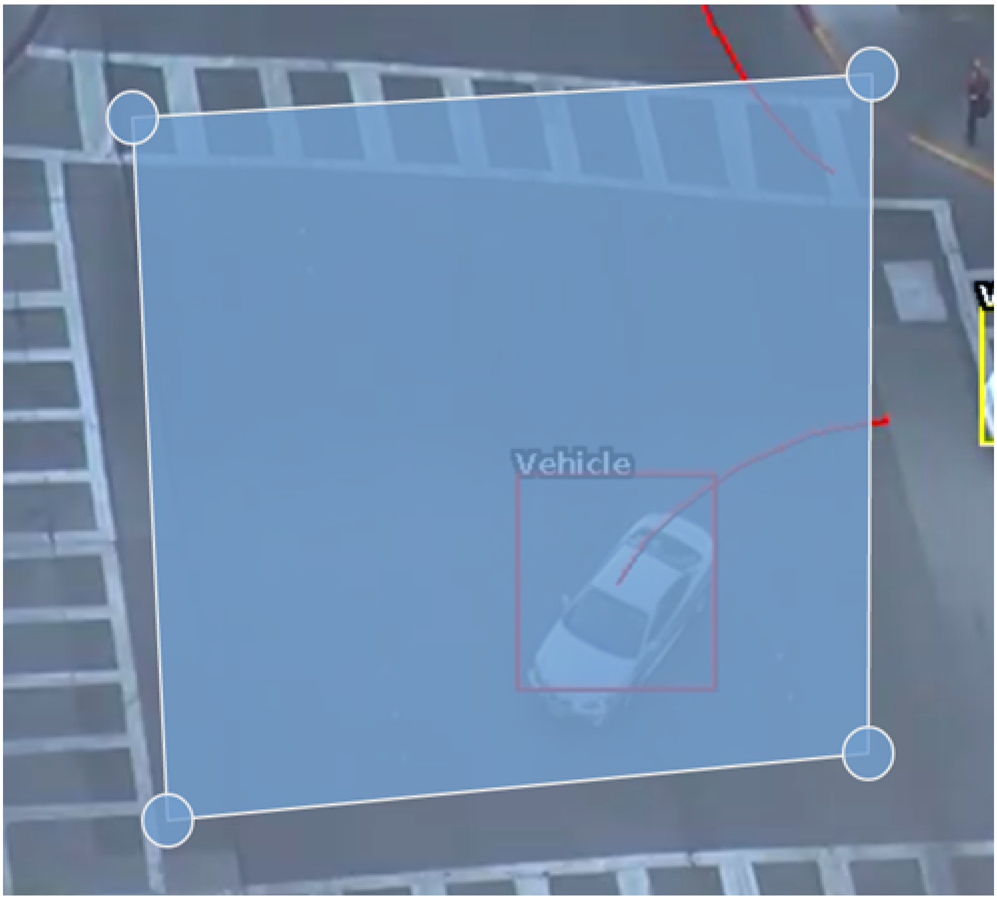
Configuration Dwell

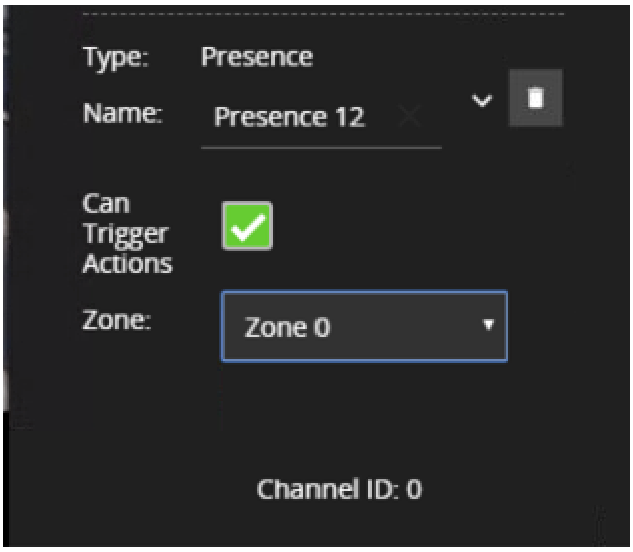
Property	Description	Default Value
Name	A user-specified name for this rule	"Direction #"
Can Trigger Actions	Specifies whether events generated by this rule trigger actions	Active
Zone	The zone this rule is associated with	None
Interval	Period of time in seconds)	1

Top Previous Next

7.7. 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

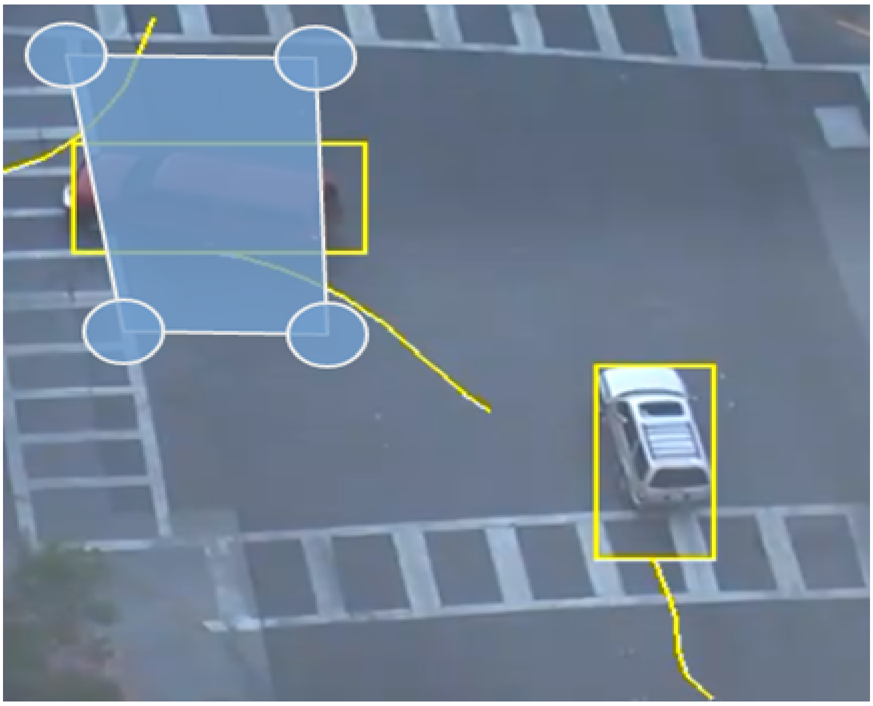
Configuration Presence

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

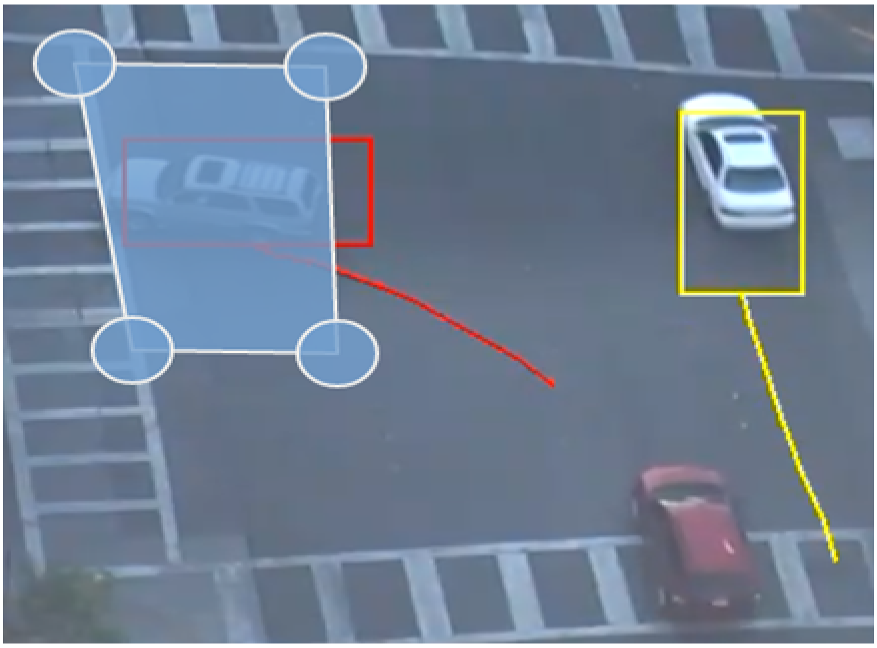
Top Previous Next

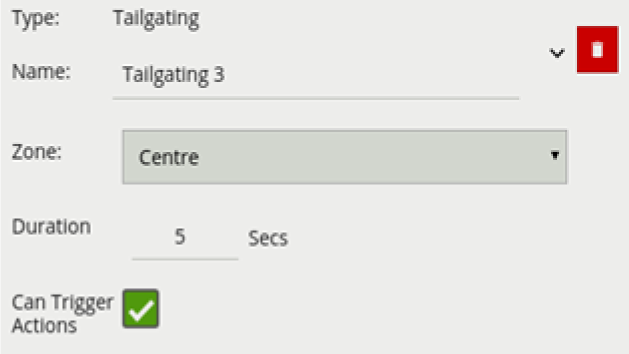
7.8. Tailgating

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	☑

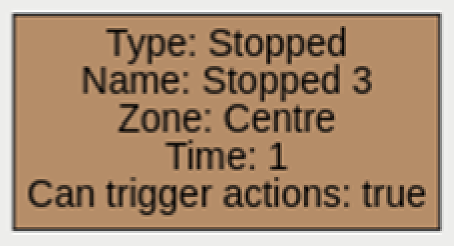
Property	Description	Default Value
Name	A user-specified name for this rule	"Tailgating #"
Zone	The zone this rule is associated with	None
Duration	The maximum amount of time between first and second object entering a zone to trigger the rule	0
Can Trigger Actions	Specifies whether events generated by this rule trigger actions	Active

Top Previous Next

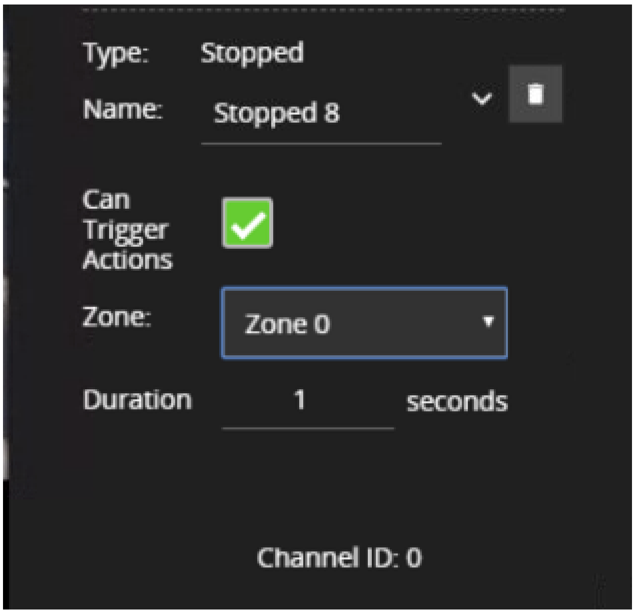
7.9. Stopped

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

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

Top Previous Next

7.10. Counting Line

A counting line is a detection filter optimised 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.

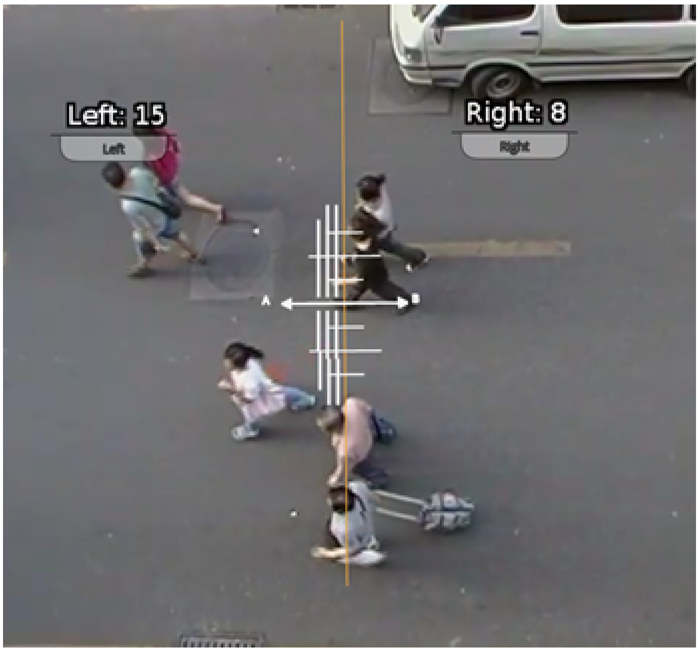
In some scenes, such as entrances with cameras installed overhead, the counting line typically will generate a higher accuracy count than using the aforementioned counters connected to a presence rule.

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.

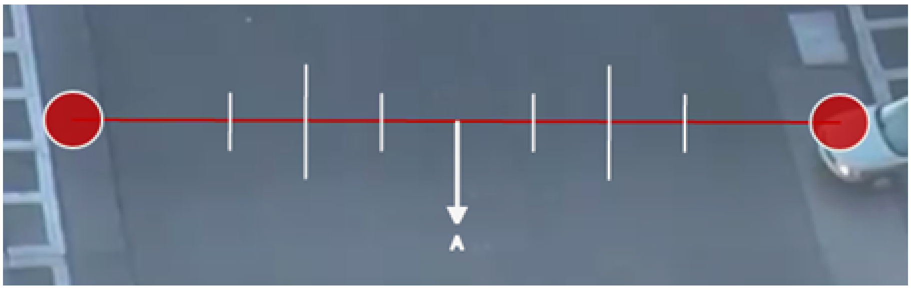


Top Previous Next

7.10.1. 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 centre of the line represent the width of the expected object. This allows the counting line to reject noise and also count multiple objects.



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.

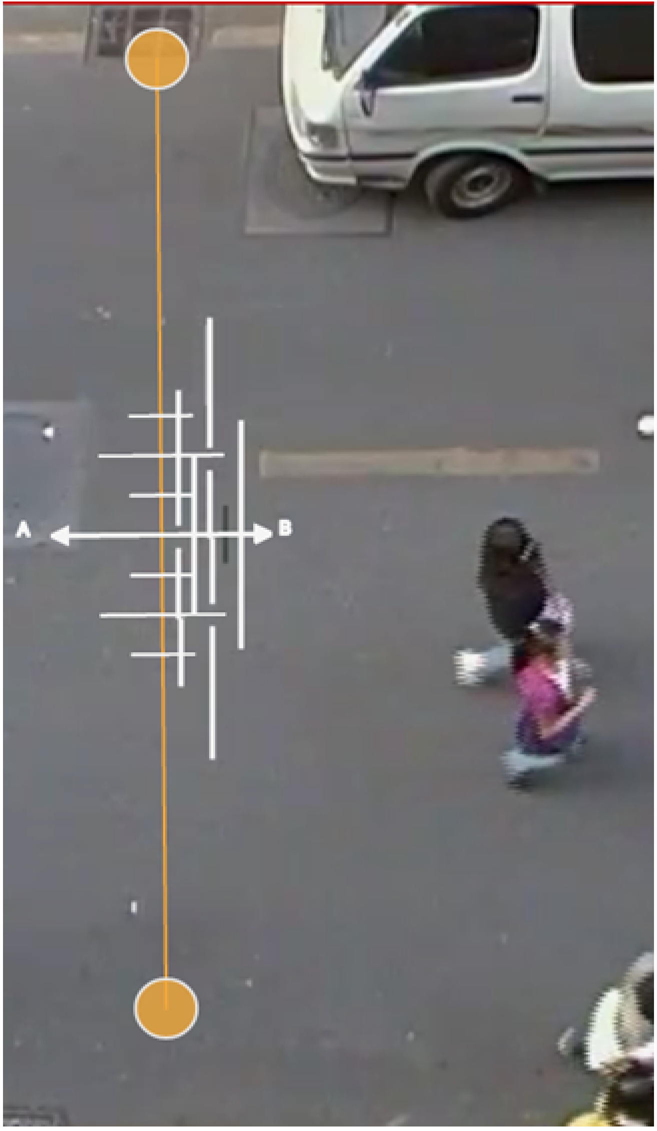
NOTE: if the Width slider is set to zero then the Enable width calibration checkbox is automatically disabled.

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.

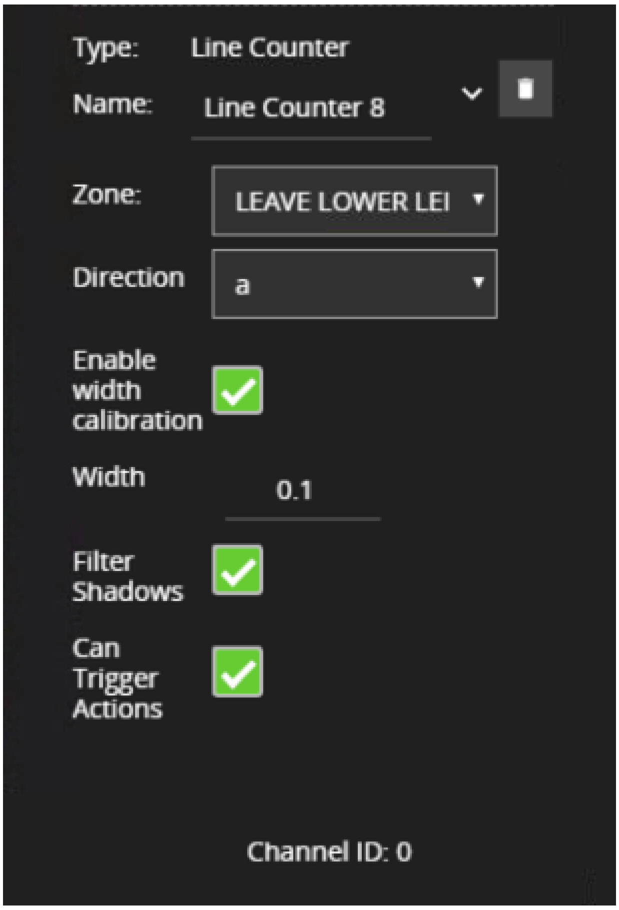
The calibration feedback is rendered as black and white lines on either side of the counting line on the Zones configurations page.
Each line represents an object detected by the counting algorithm.
The width of the line shows the width of the object detected by the line.
The last few detections are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



Top Previous Next

7.10.2. 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

Top Previous Next

7.11. 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.

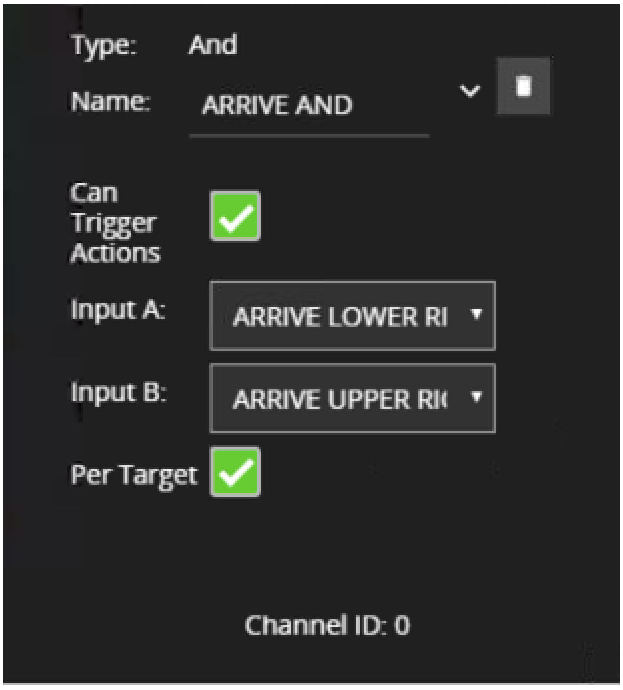
The complete list of conditional rules are:

- And
- Continuously
- Counter
- Or
- Previous

Top Previous Next

7.11.1. 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.

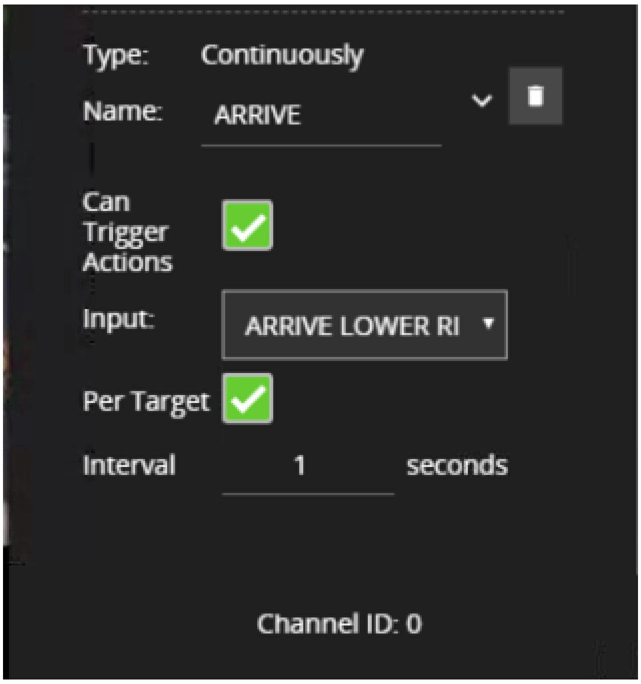
State	Per Target	Outcome
Object A in Input A, Object B in input B	On	Two events were generated, one for each object
Object A in Input A, Object B in input B	Off	Only one event generated

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.

Top Previous Next

7.11.2. 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.

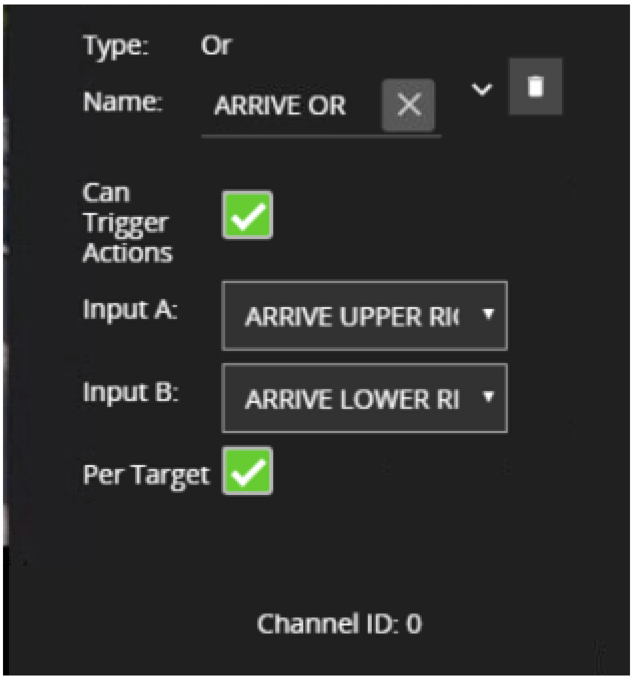
When Per Target is off, the state will change when:

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

Top Previous Next

7.11.3. 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.

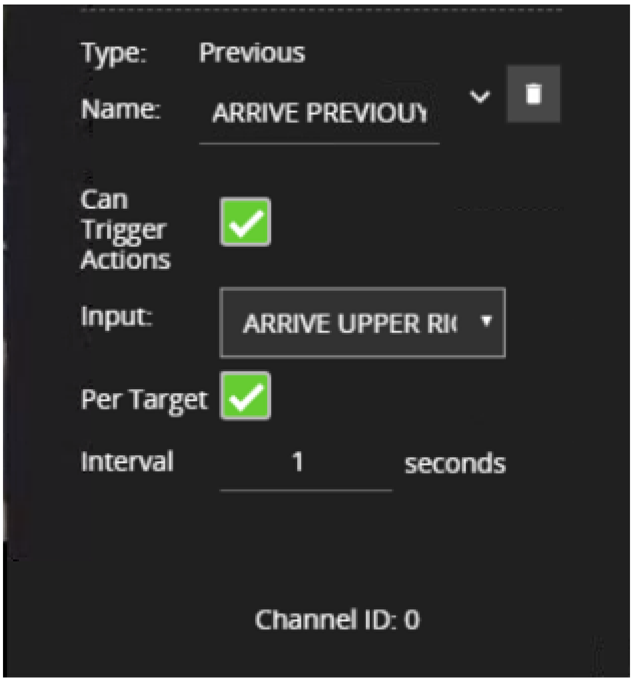
State	Per Target	Outcome

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

Top Previous Next

7.11.4. 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 PREVIOU

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

Top Previous Next

7.12. Filters

A filter cannot trigger an action on its own as it requires another basic input, filter or conditional rule to trigger.
An example of this is the Object rule.

The complete list of filters are:

1. Speed Filter
2. Object Filter
3. Colour Filter
4. Deep Learning Filter

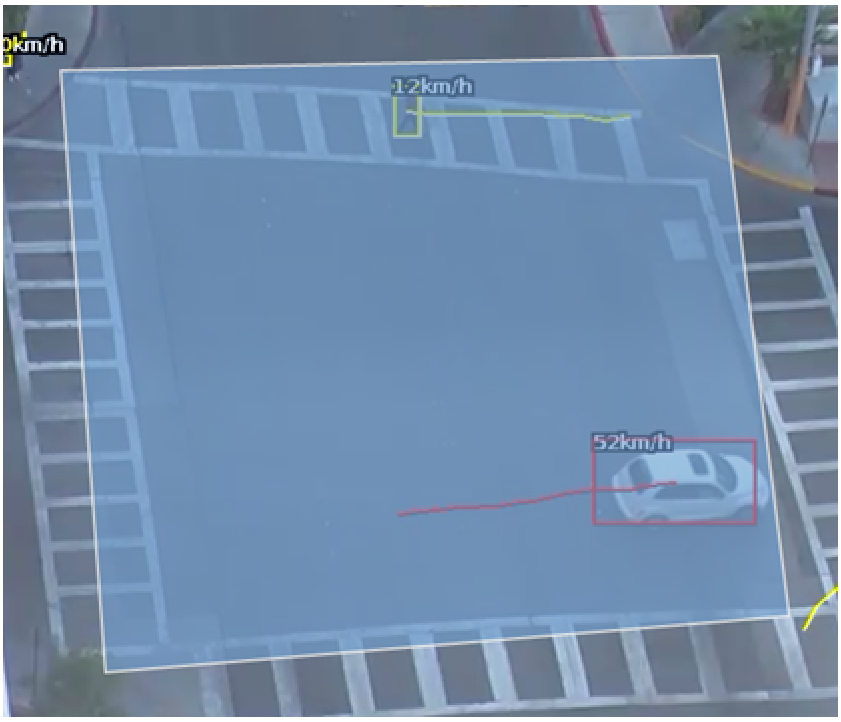
Due to the nature of the deep learning algorithm which powers the Deep Learning Filter, it can not be used as an input to another filter or logical rule.

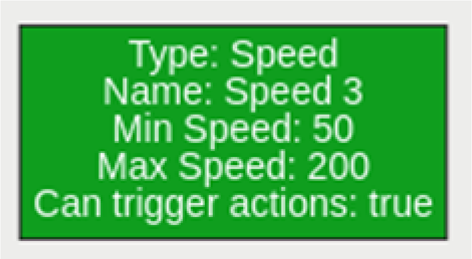
Top Previous Next

7.12.1. Speed Filter

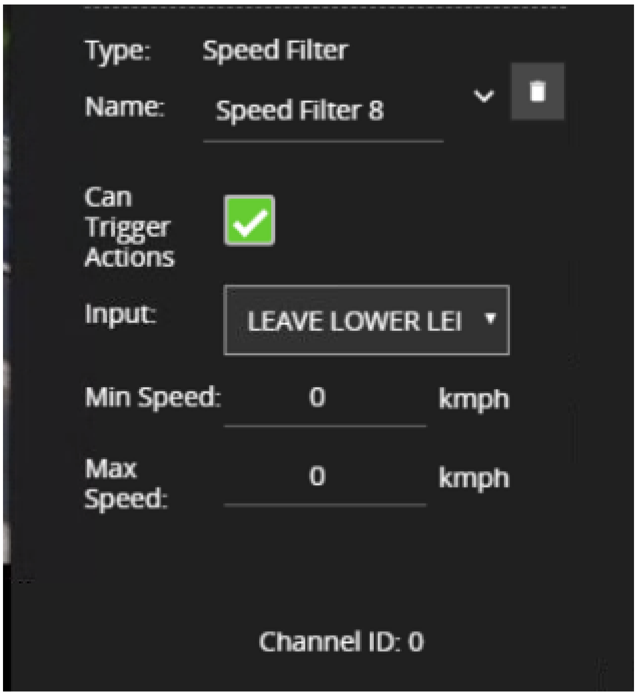
The speed filter provides a way to check if the speed of an object which has triggered an input is moving within the range of speeds defined by a lower and upper boundary.
Note: The channel must be calibrated in order for the speed filter to be available.
Commonly this rule is combined with a presence 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

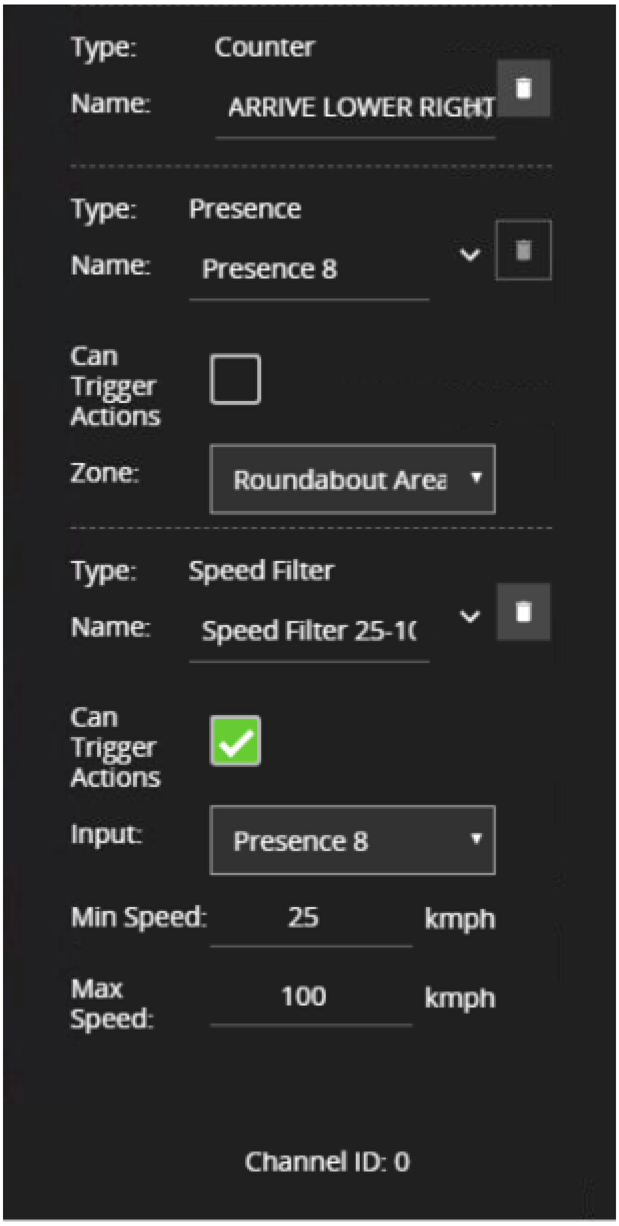
Property	Description	Default Value
Name	A user-specified name for this rule	"Speed #"
Can Trigger Actions	Specifies whether events generated by this rule trigger actions	Active
Input	The input rule	None
Min Speed	The minimum speed (km/h) an object must be going to trigger the rule	0
Max Speed	The maximum speed (km/h) an object can be going to trigger the rule	0

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

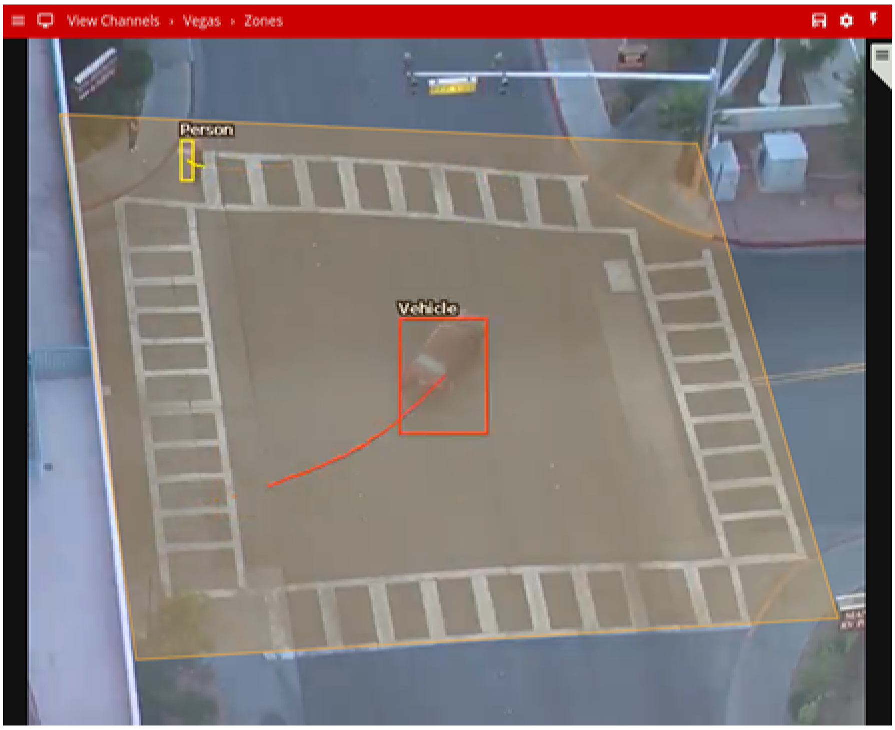
Type: Speed Filter
Name: Speed Filter 25-1(
Can Trigger Actions ☑
Input: Presence 8
Min Speed: 25 kmph
Max Speed: 100 kmph

Channel ID: 0

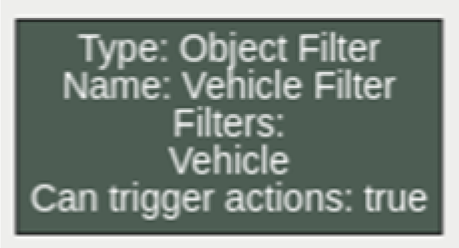
Top Previous Next

7.12.2. Object Filter

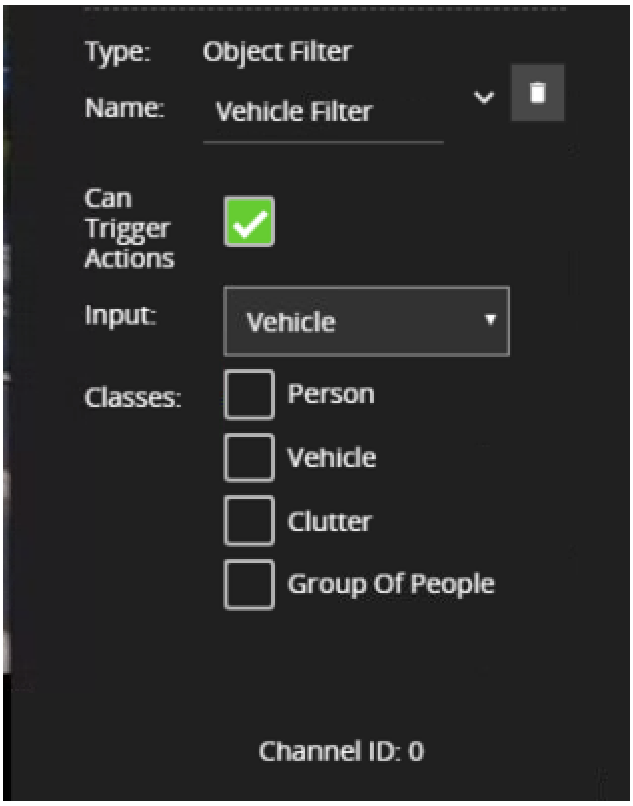
The object classification filter provides the ability to filter out objects, which trigger a rule if they are not classified as a certain class (e.g. person, vehicle).
The object classification filter must be combined with another rule(s) to prevent unwanted objects from triggering an alert, an example 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.
Note: 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

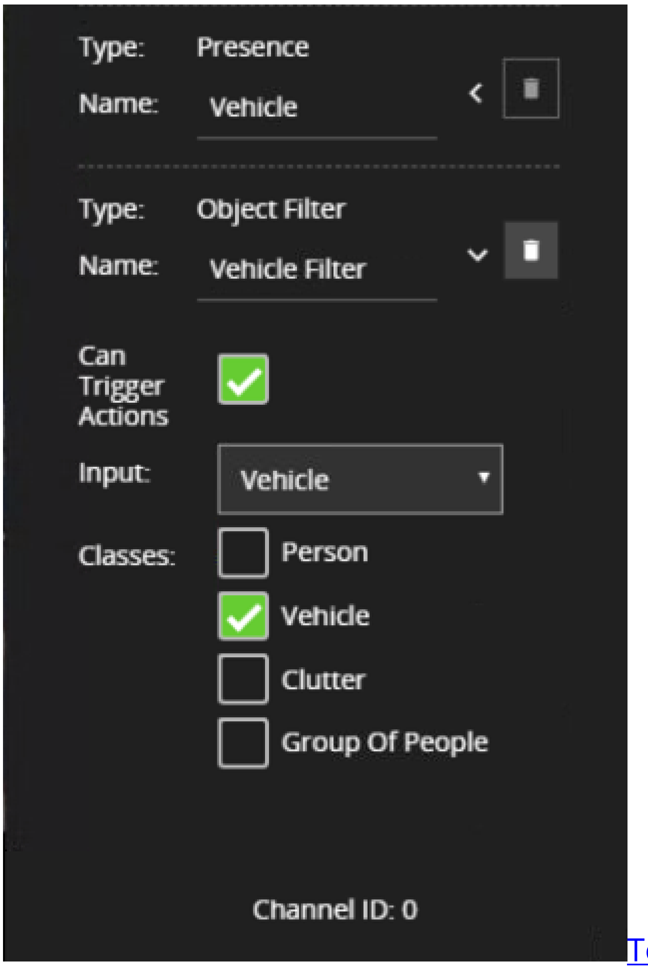
Classes:
□ Person
□ Vehicle
□ Clutter
□ Group Of People

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

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

Top Previous Next

7.12.3. Colour Filter

The colour filter utilises 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.

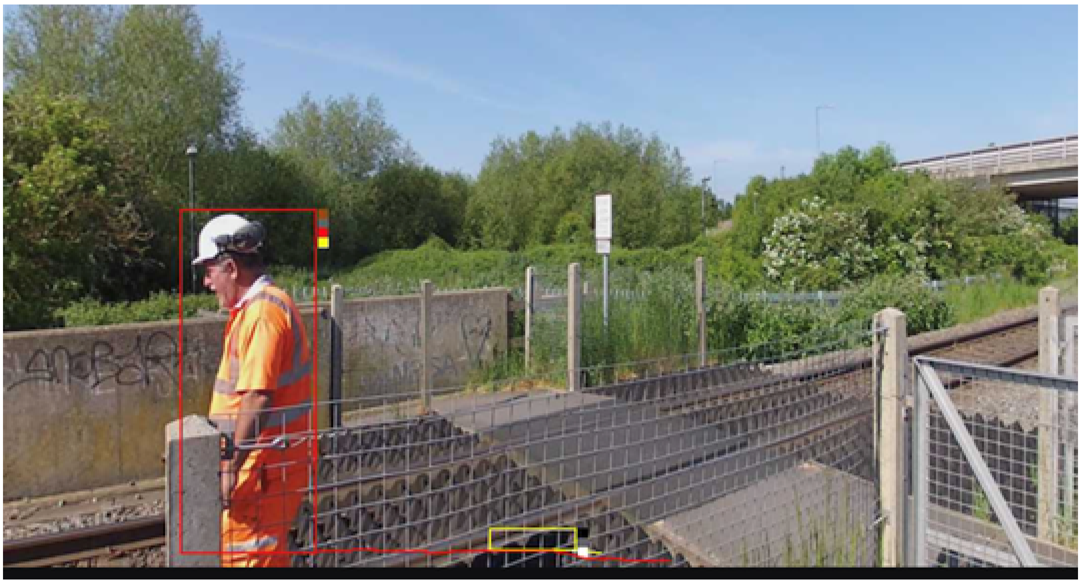
The colour filter allows you to select one or more of these colour bins and will trigger if the subject-object is made up of one or more of those selected colours.

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.

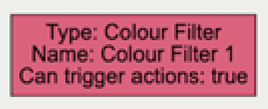
Typically, the colour classification filter would be combined with another rule(s) to prevent unwanted objects from triggering an alert, an example rule graph is provided to illustrate this below.



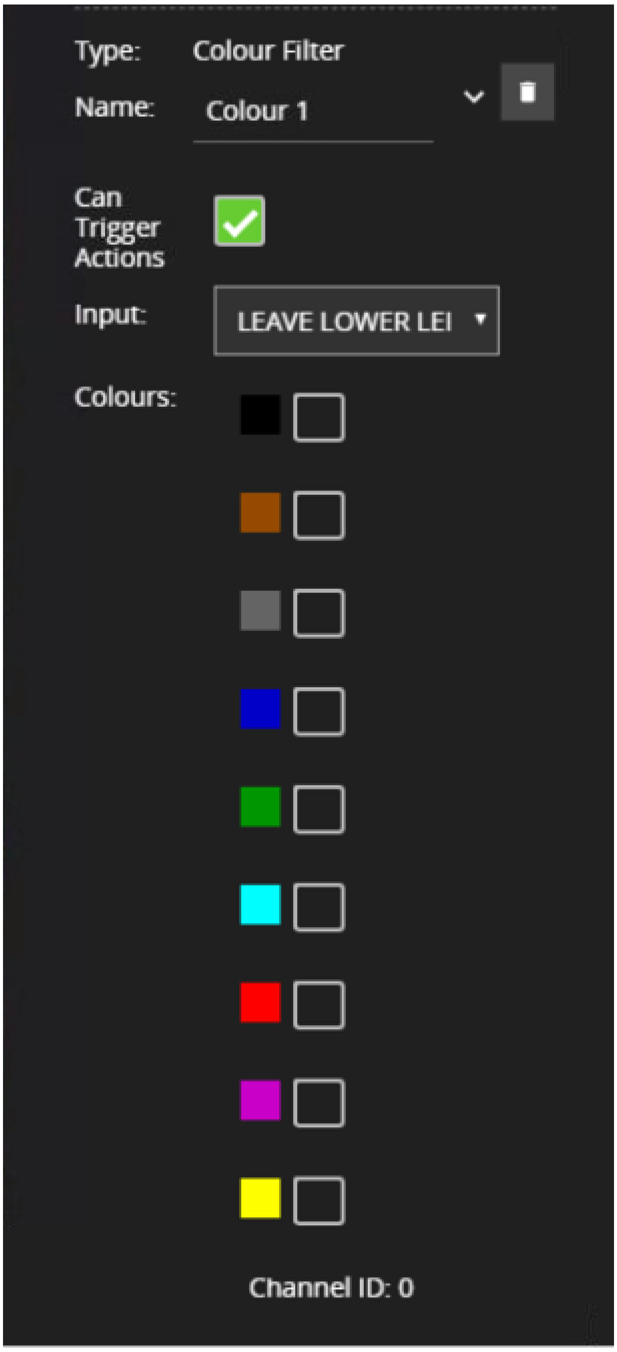
The previous image illustrates how the object classification filter configured with Vehicle class, includes only Vehicle objects.

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

Note: 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:

⬛ □
🟫 □
▨ □
🟦 □
🟩 □
▨ □
🟥 □
🟪 □
🟨 □

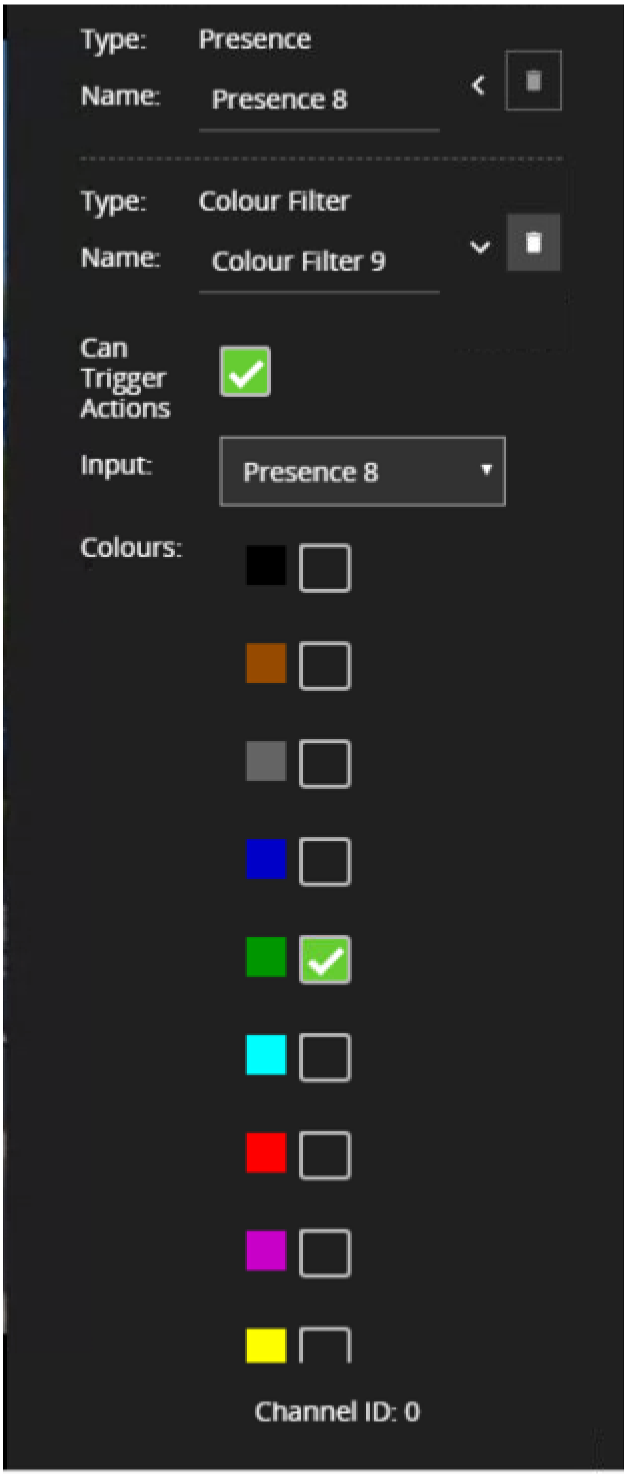
Channel ID: 0

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
Input	The input rule	None
Colours	The colours allowed to trigger an alert	All Unchecked

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 Yellow 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:
⬛ □
🟫 □
◼ □
🟦 □
🟩 ☑
◼ □
🟥 □
🟪 □
🟨 □

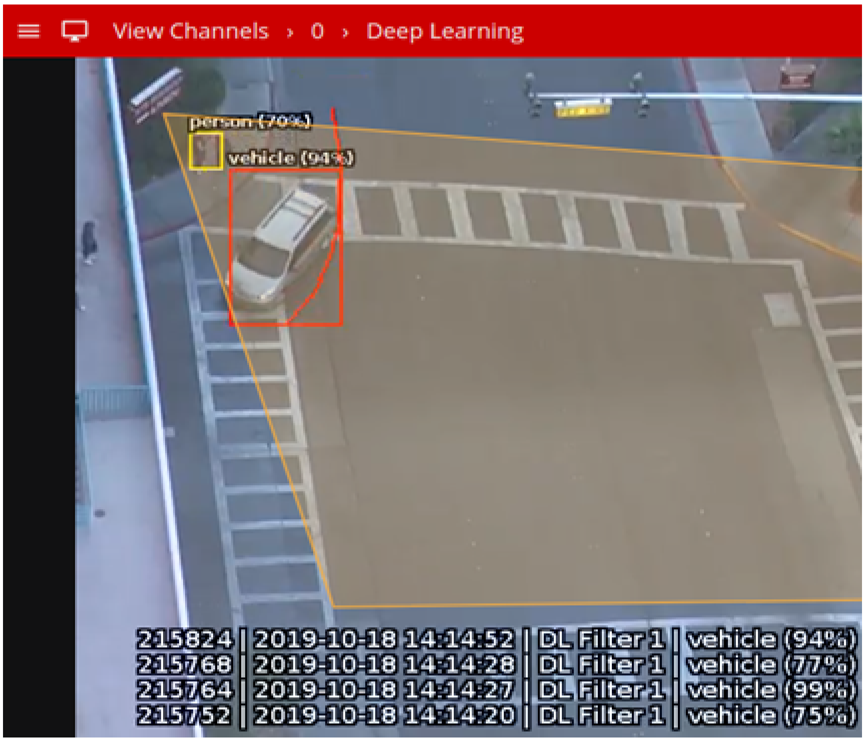
Channel ID: 0

Top Previous Next

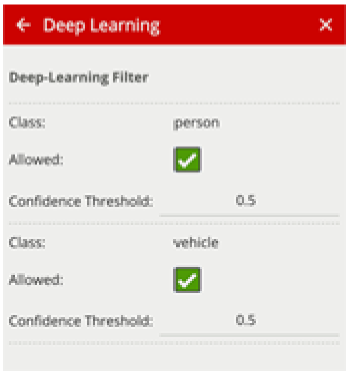
7.12.4. Deep Learning Filter

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



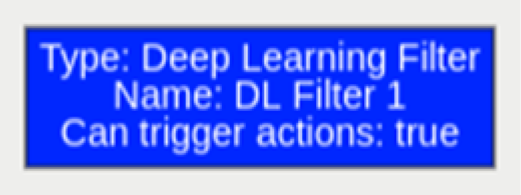
```
215824 | 2019-10-18 14:14:52 | DL Filter 1 | vehicle (94%)
215768 | 2019-10-18 14:14:28 | DL Filter 1 | vehicle (77%)
215764 | 2019-10-18 14:14:27 | DL Filter 1 | vehicle (99%)
215752 | 2019-10-18 14:14:20 | DL Filter 1 | vehicle (75%)
```



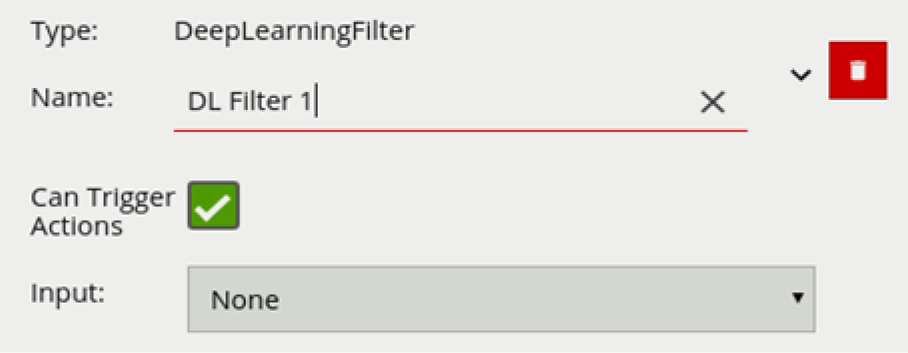
Deep-Learning Filter

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

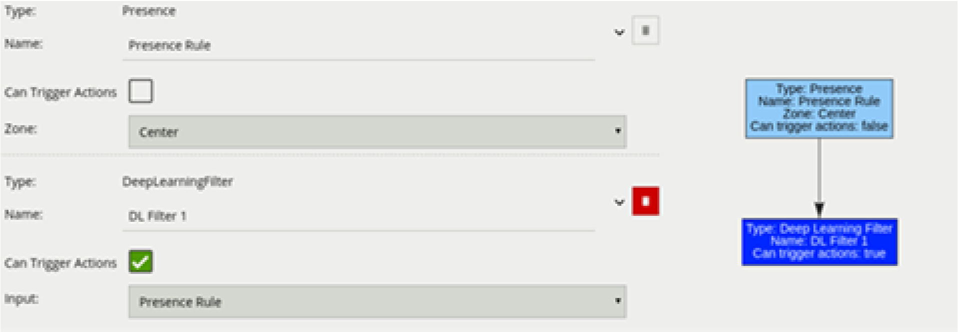
Can Trigger Actions ☑

Input: None

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

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 ☑

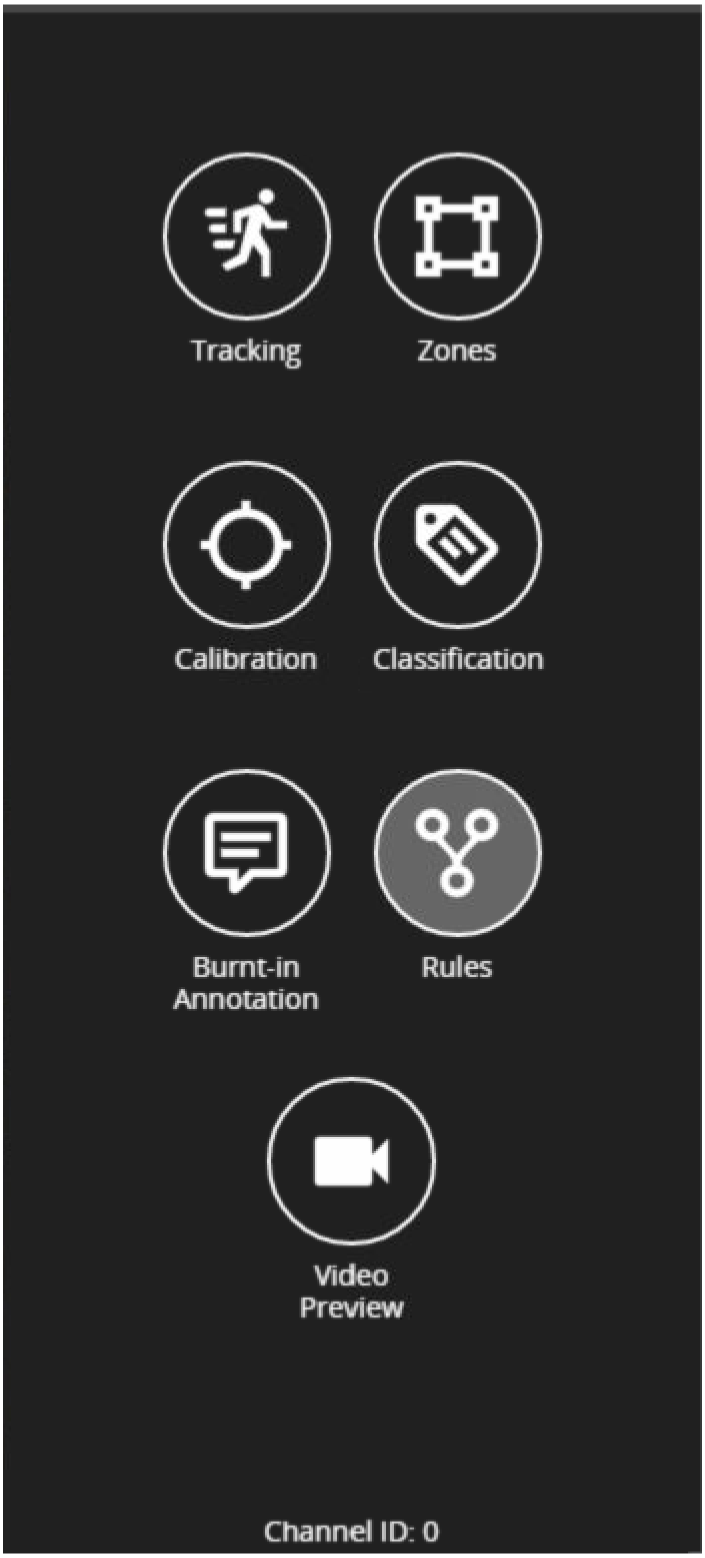
Input: Presence Rule

Type: Presence
Name: Presence Rule
Zone: Center
Can trigger actions: false
Type: Deep Learning Filter
Name: DL Filter 1
Can trigger actions: true

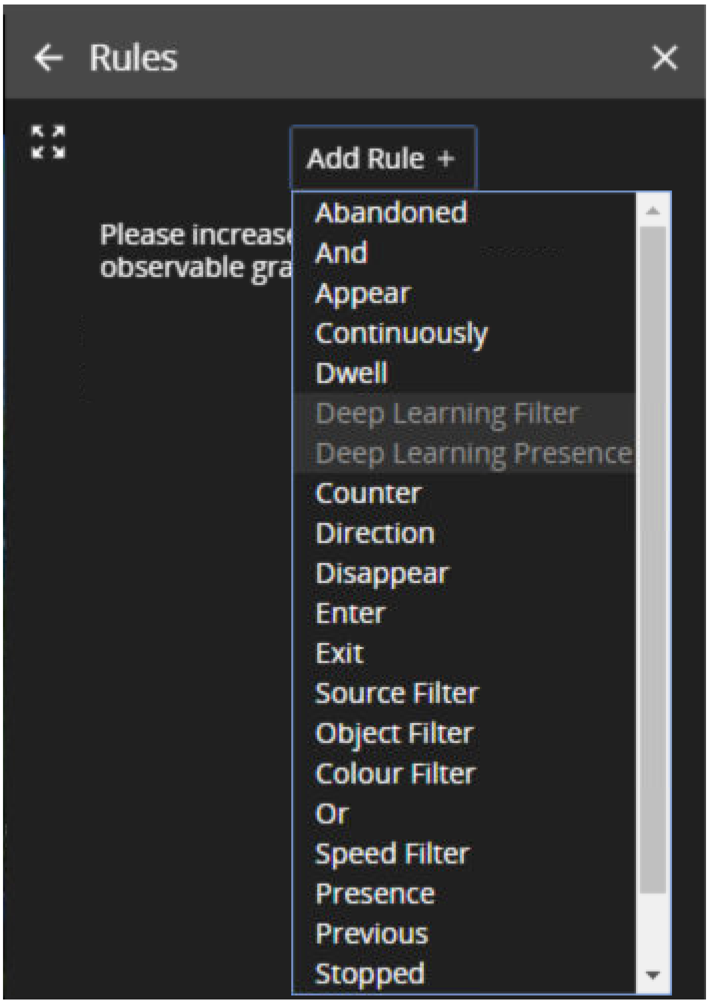
Top Previous Next

7.13. How to create rules

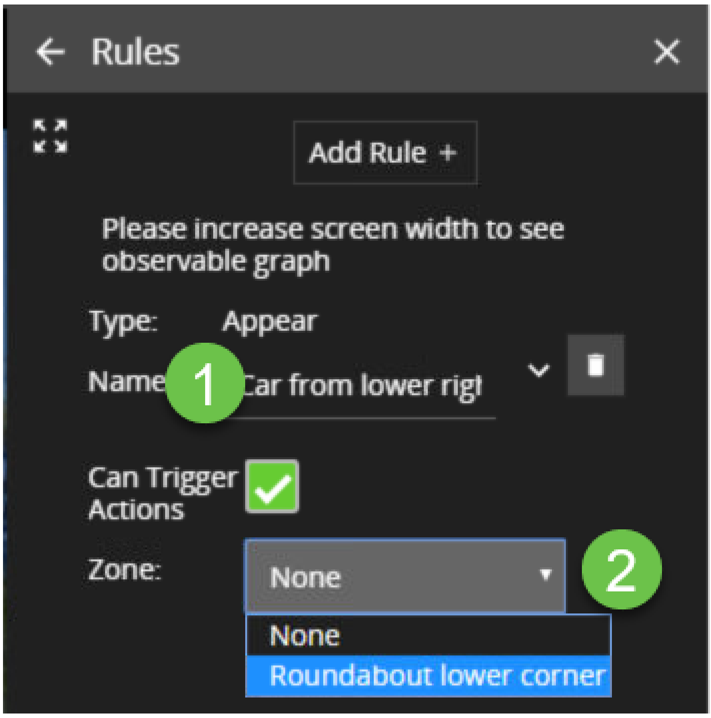
1. Click **Rules**



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



1. Set the name for the rule
2. Select zone



Please increase screen width to see observable graph

Type: Appear

Name: Car from lower rigl

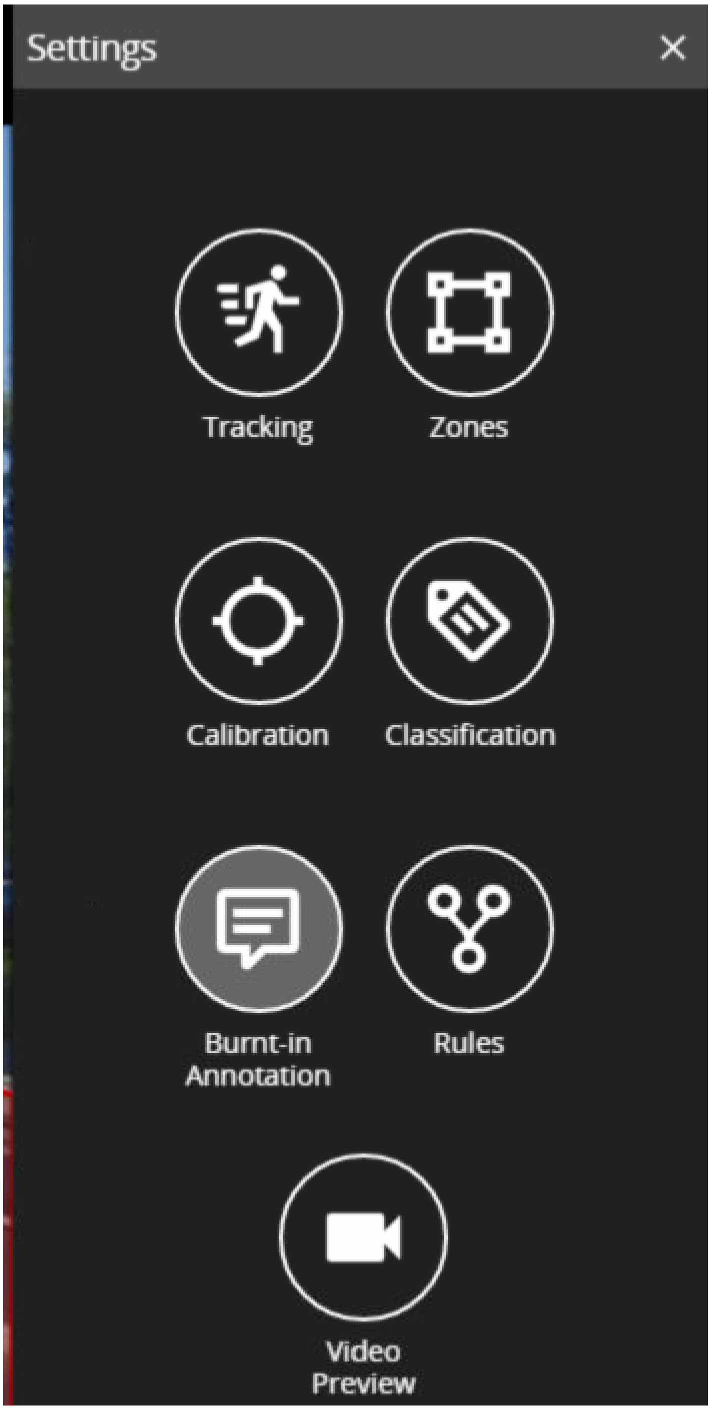
Can Trigger Actions ☑

Zone: None

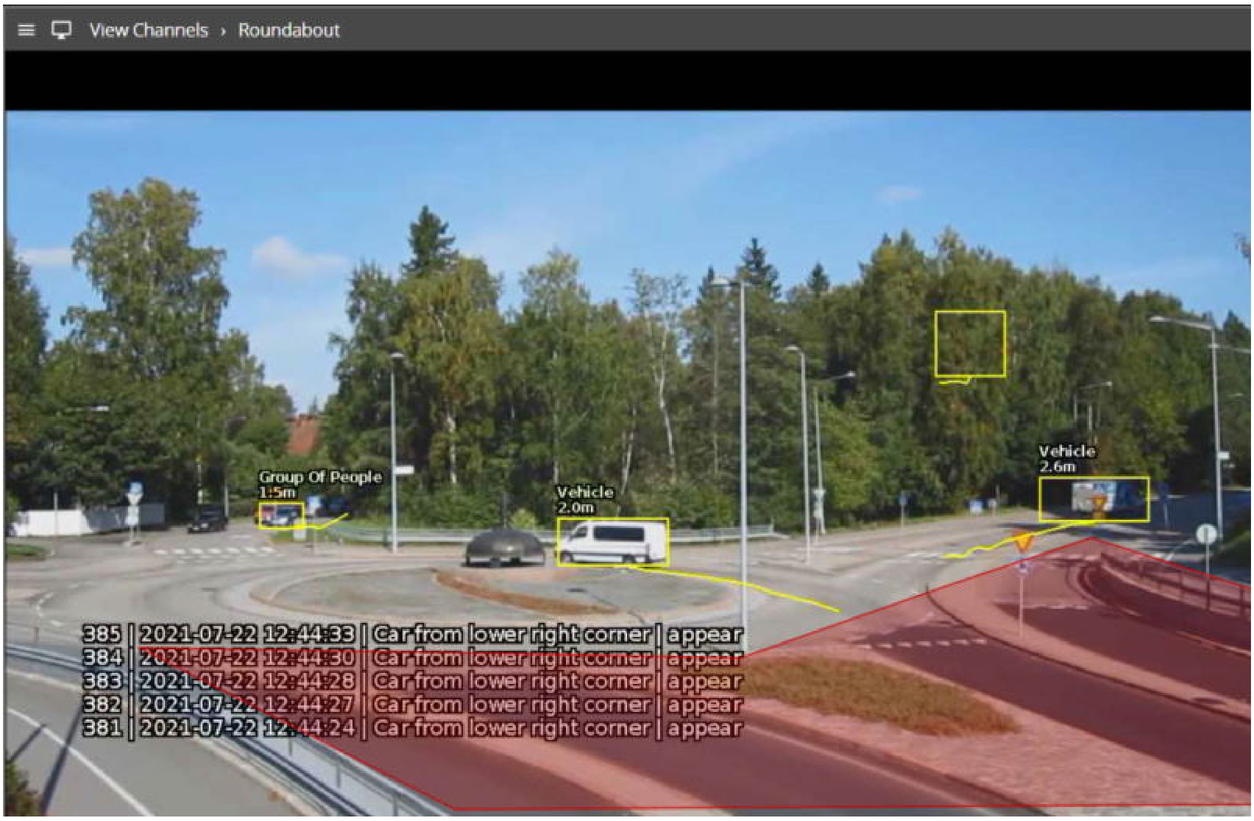
Top Previous Next

7.14. How to test rules

Open Burnt-in Annotation



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
```

Top Previous Next

7.15. Counters

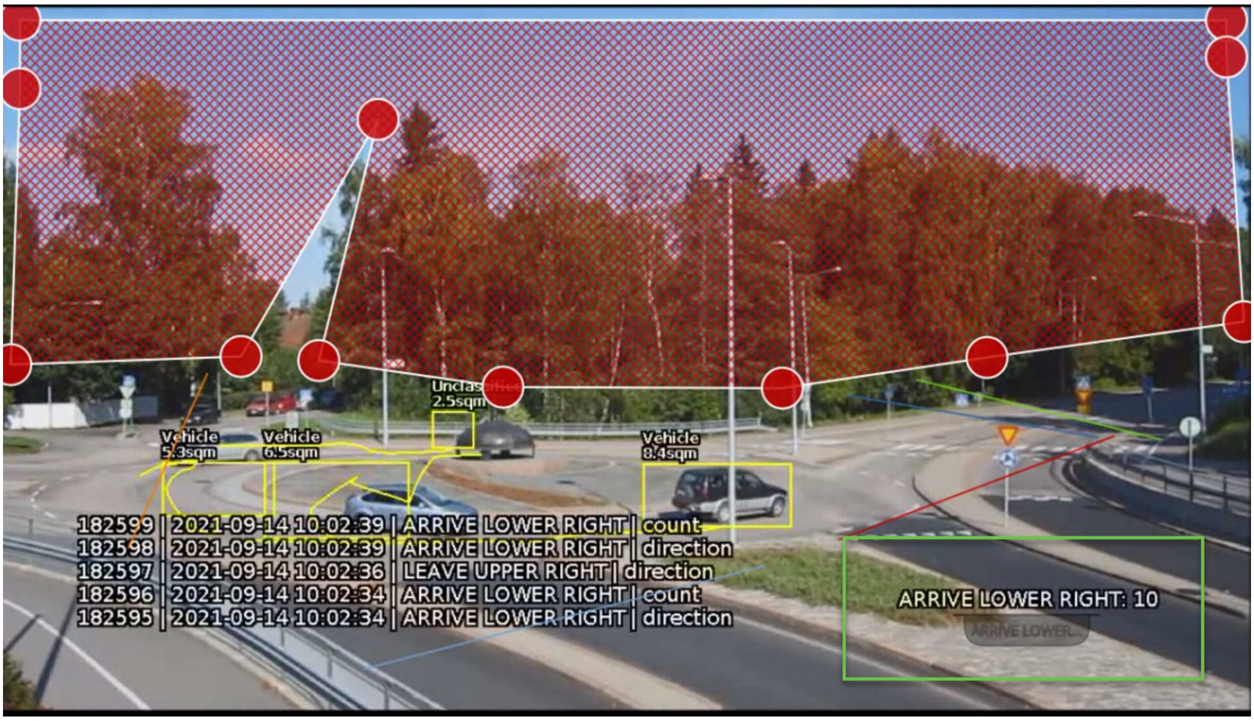
These counters are only visible in the VCA configuration. To use counters in the Spotter please refer to the Spotter manual.
Counters can be configured to count the number of times a rule is triggered, for example, the number of people crossing a line.

The counter rule is designed to be 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.

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

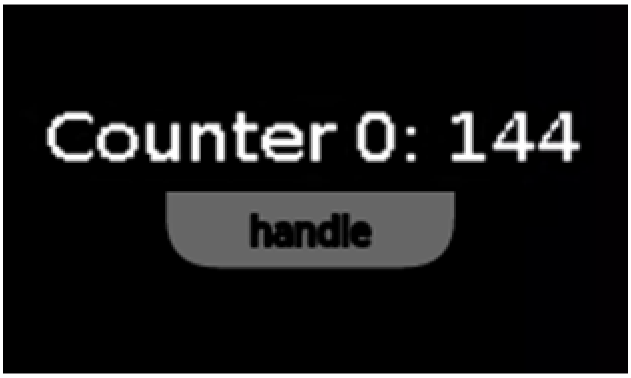


```
182599 | 2021-09-14 10:02:39 | ARRIVE LOWER RIGHT | count
182598 | 2021-09-14 10:02:39 | ARRIVE LOWER RIGHT | direction
182597 | 2021-09-14 10:02:36 | LEAVE UPPER RIGHT | direction
182596 | 2021-09-14 10:02:34 | ARRIVE LOWER RIGHT | count
182595 | 2021-09-14 10:02:34 | ARRIVE LOWER RIGHT | direction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

ARRIVE LOWER RIGHT: 10

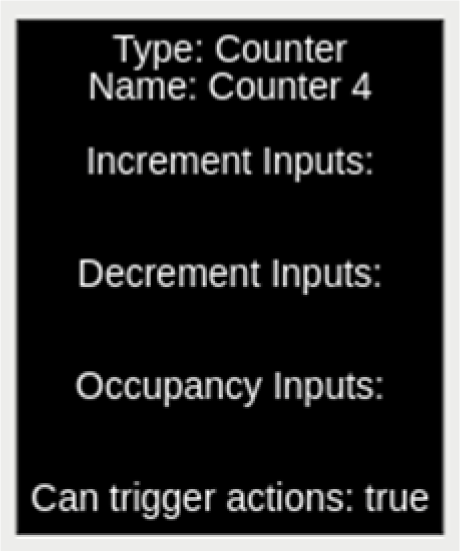
Positioning Counters

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.



Counter 0: 144

handle



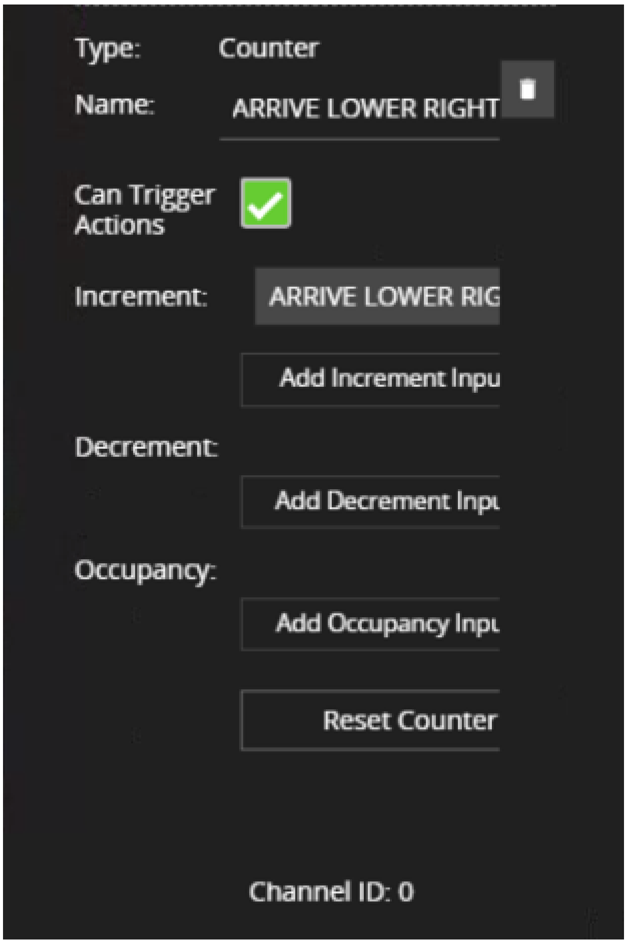
Type: Counter
Name: Counter 4

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

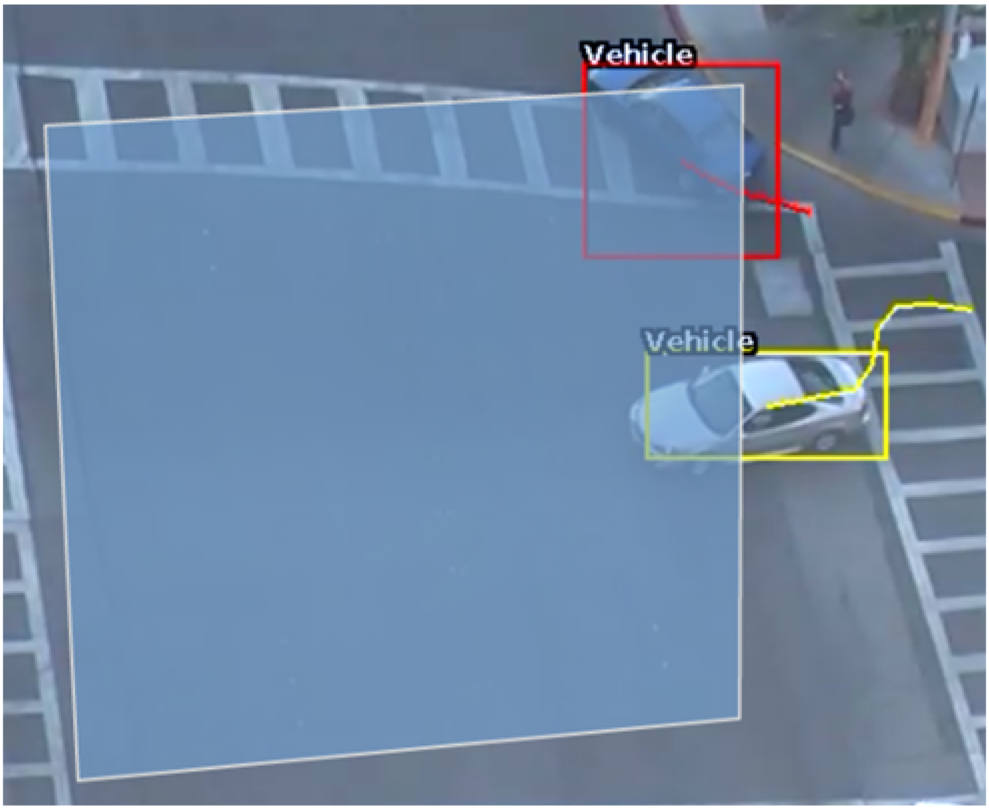
Reset Counter	A button allowing the counter value to be reset to 0	None

* 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'.

Top Previous Next

7.16. 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.



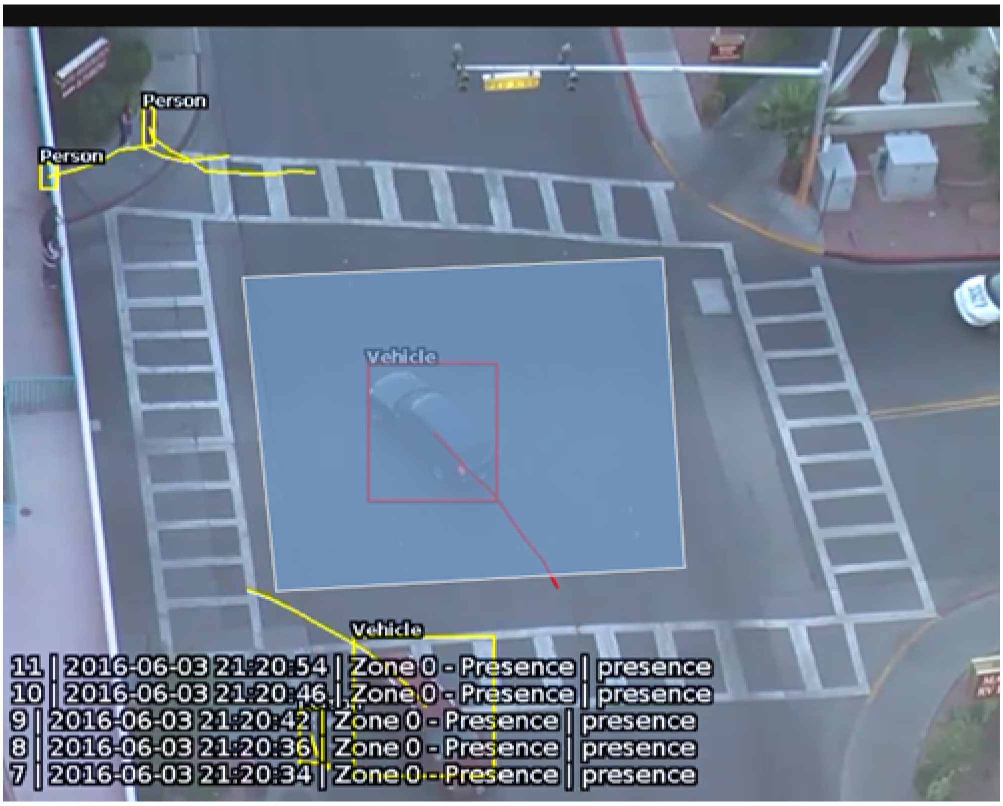
Top Previous Next

7.17. 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.

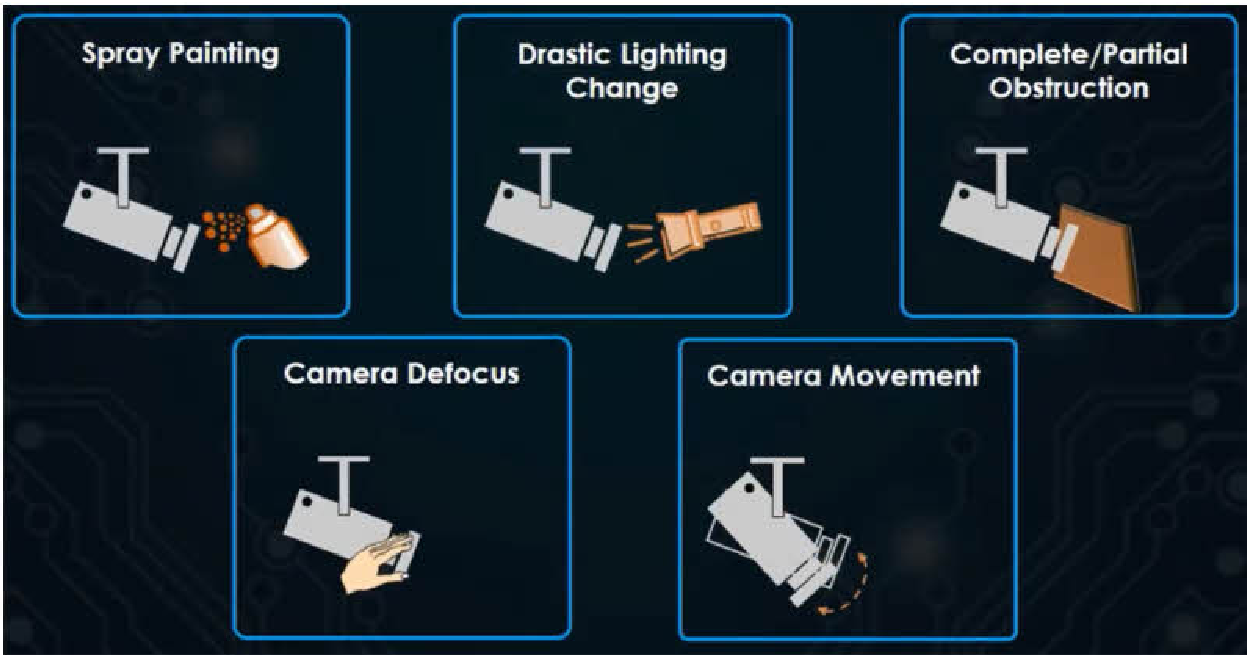
As seen below, when an event is raised, the default settings render details of the event in the lower half of the video stream.
Object class annotations in this example are generated through



```
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
```

Top Previous Next

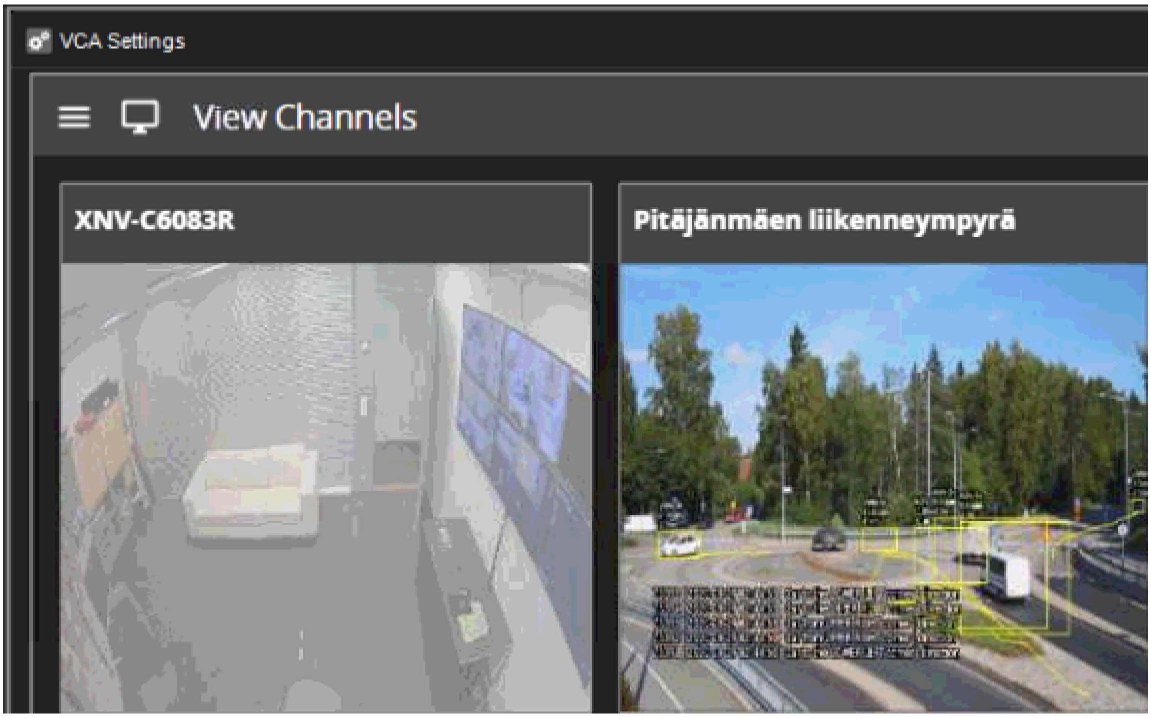
8. Tamper Detection



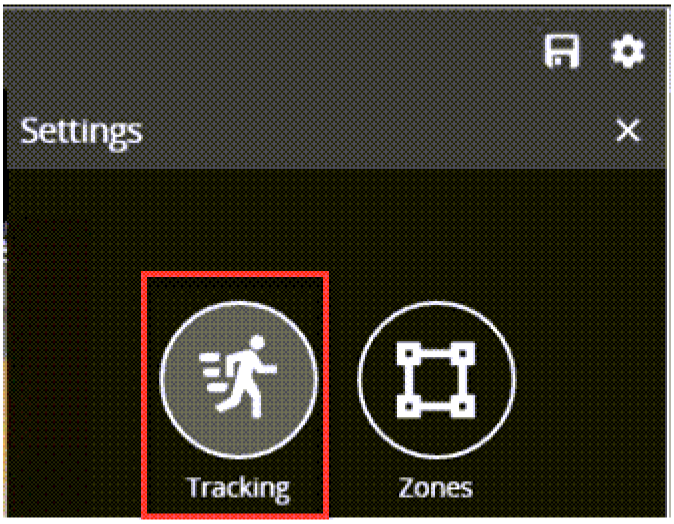
Previous Next

8.1. How to enable Tamper Detection

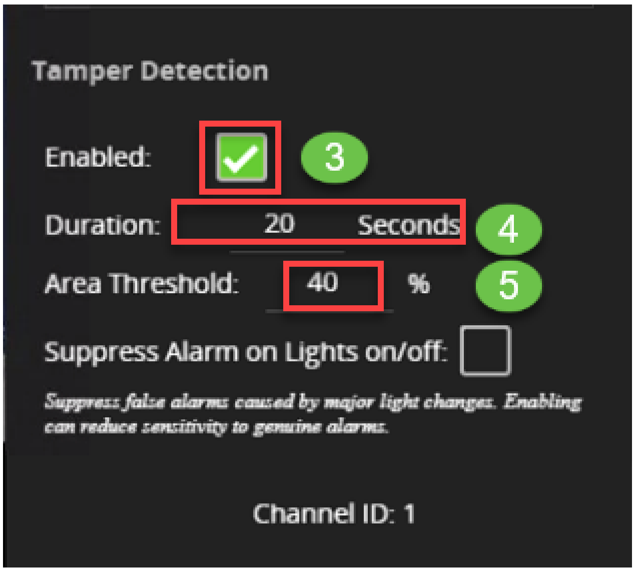
1. Open camera from the View Channels



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: □

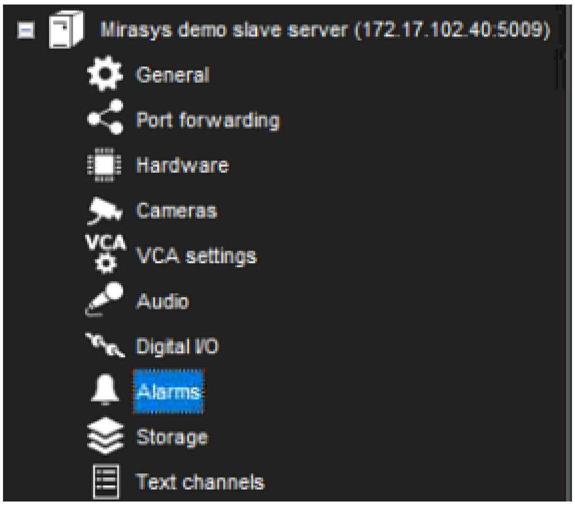
Suppress false alarms caused by major light changes. Enabling can reduce sensitivity to genuine alarms.

Channel ID: 1

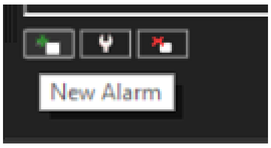
Top Previous Next

8.2. How to create an alarm from Tamper Detection

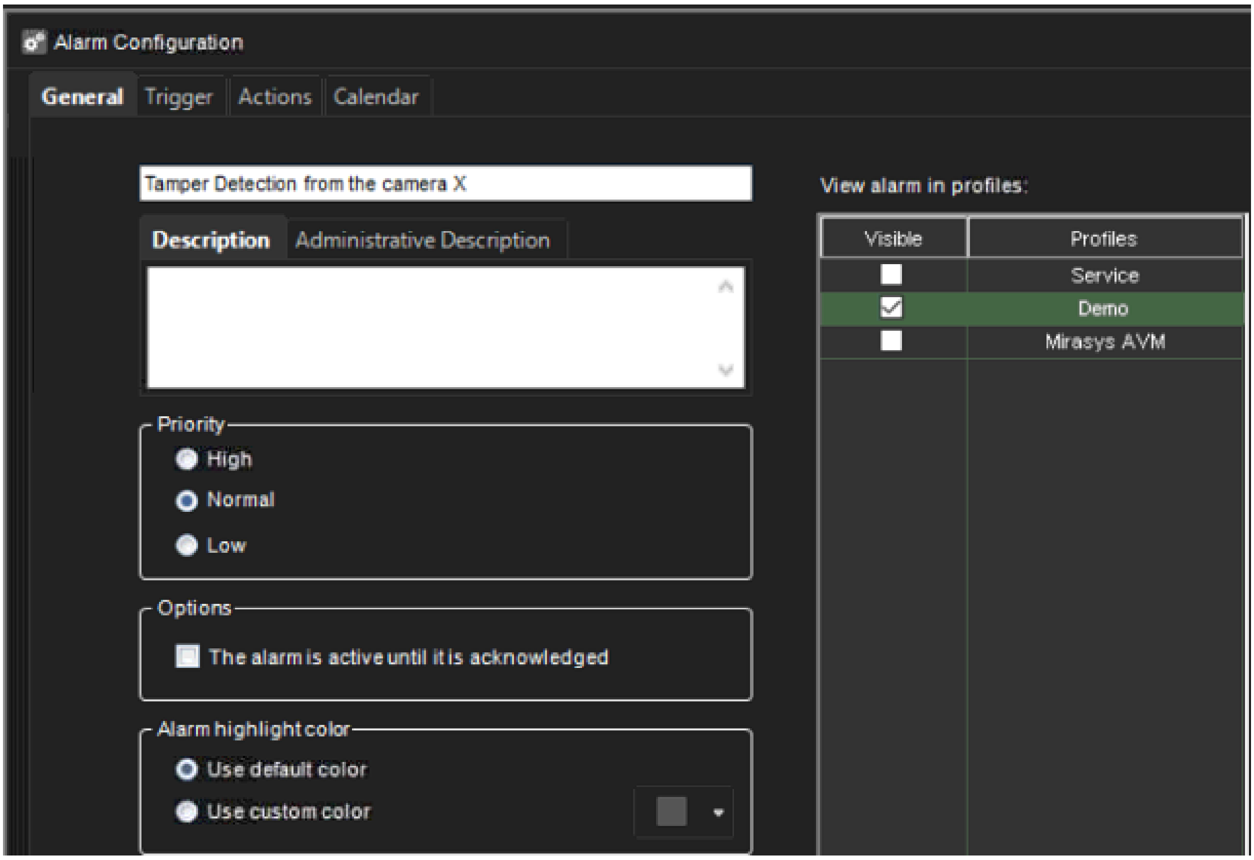
1. Open Alarms from the needed VMS server



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

Priority
- ○ High
- ● Normal
- ○ Low

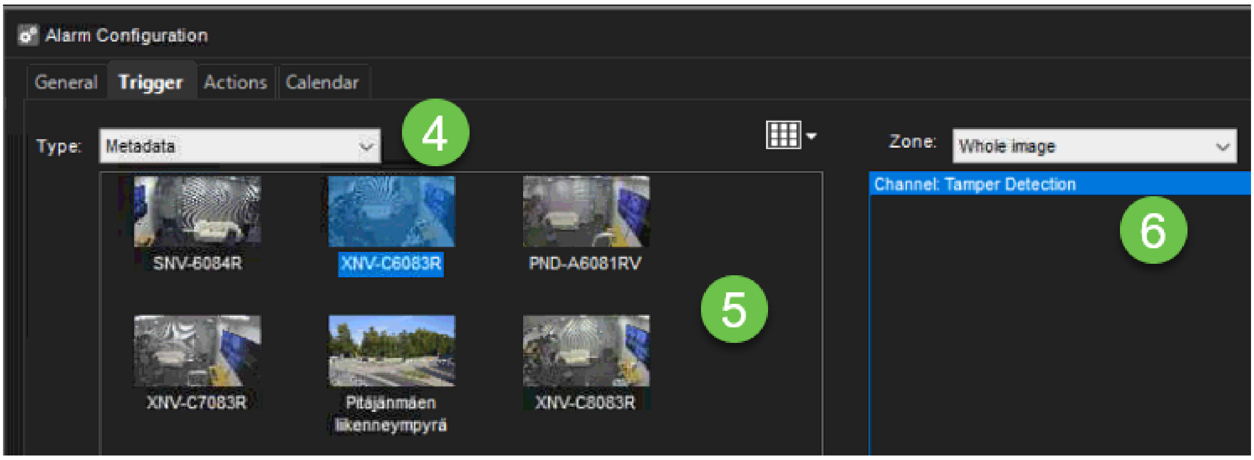
Options
- □ The alarm is active until it is acknowledged

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

View alarm in profiles:

Visible	Profiles
□	Service
☑	Demo
□	Mirasys AVM

4. Select Trigger **Metadata**
5. Select camera from the list
6. Select Tamper Detection



Alarm Configuration

General | Trigger | Actions | Calendar

Type: Metadata

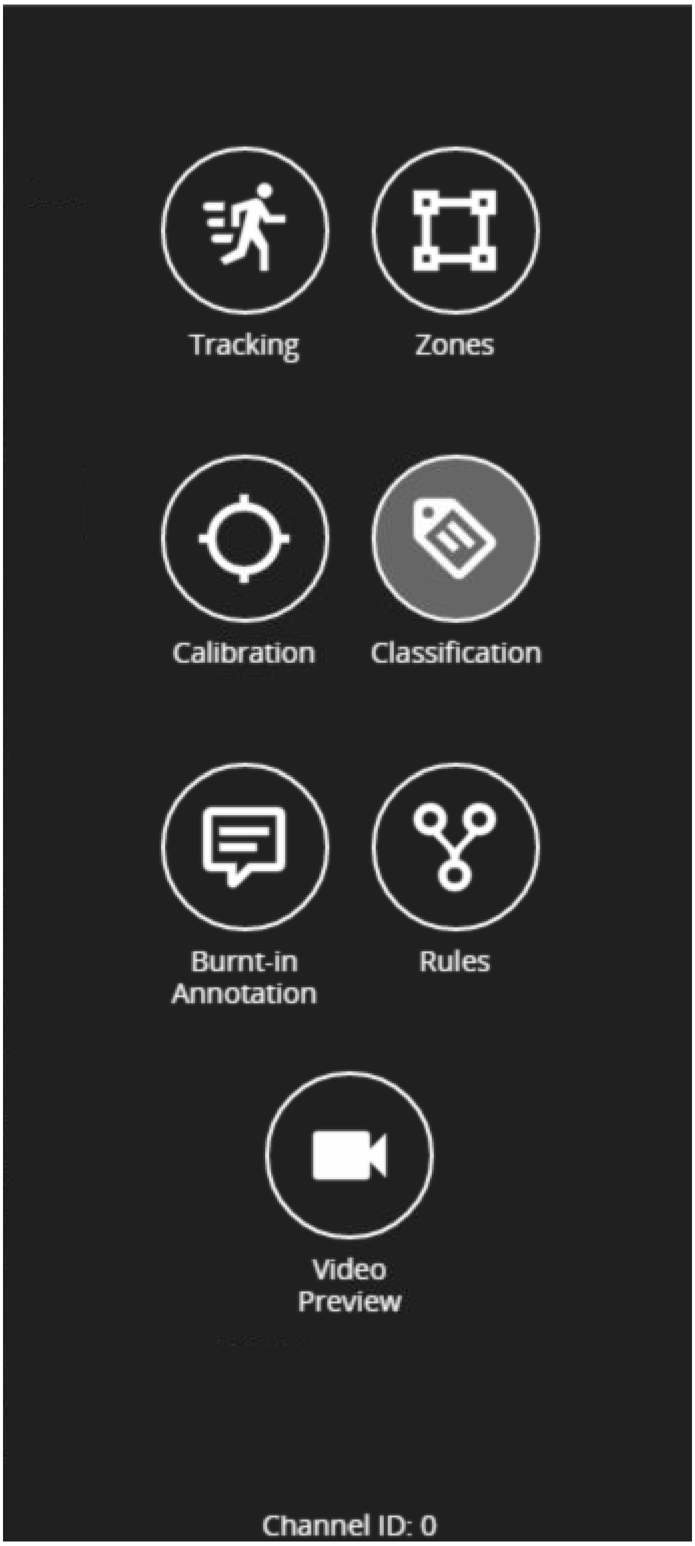
Zone: Whole image

Channel: Tamper Detection

[Top](#) [Previous](#) [Next](#)

9. 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.



Previous Next

9.1. 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.
VCAcore's object classification performs classification by comparing these properties to a set of configurable object classifiers.
VCAcore comes pre-loaded with the most common object classifiers, and in most cases, these will not need to be modified.
Objects are classified according to how their calibrated properties match the classifiers.
Each classifier specifies a speed range and an area range.
Objects with properties that fall within both ranges of speed and area will be classified as being an object of the corresponding class.
Note: If multiple classes contain overlapping speed and area ranges then object classification may be ambiguous (since an object will match more than one class).
In this case, the actual classification is not specified and may be any one of the overlapping classes.
The classification data from object classification can be accessed via template tokens.

Top Previous Next

9.2. Configuration

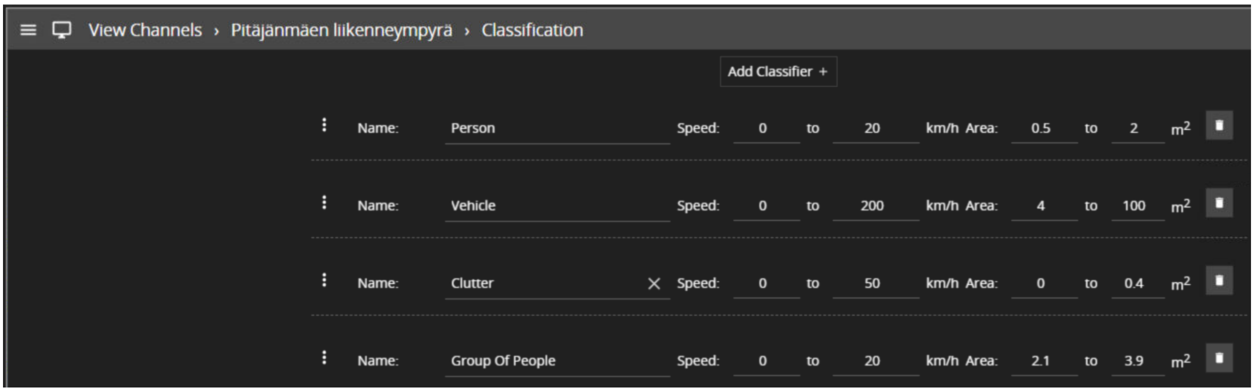
In some situations, it might be desirable to change the classifier parameters or add new object classifiers.
The classification menu can be used to make these changes.

Each of the UI elements is described below:

⋮ : Click and drag to rearrange the order of the classification groups.

1. **Name:** Specifies the name of the classification group.
2. **Speed:** Sets the speed range for the classification group. Objects which fall within the speed and area ranges will be classified with this group.
3. **Area:** Sets the area range for the classification group. Objects which fall within the speed and area ranges will be classified with this group.
4. 🗑 : Deletes the classification group.

To add a new classifier, click the Add Classifier button Add Classifier.
[Enabling Calibration](#) must be enabled on each channel object classification is to be used.
If not enabled, any rules that include an object filter will not trigger.



≡ View Channels › Pitäjänmäen liikenneympyrä › Classification

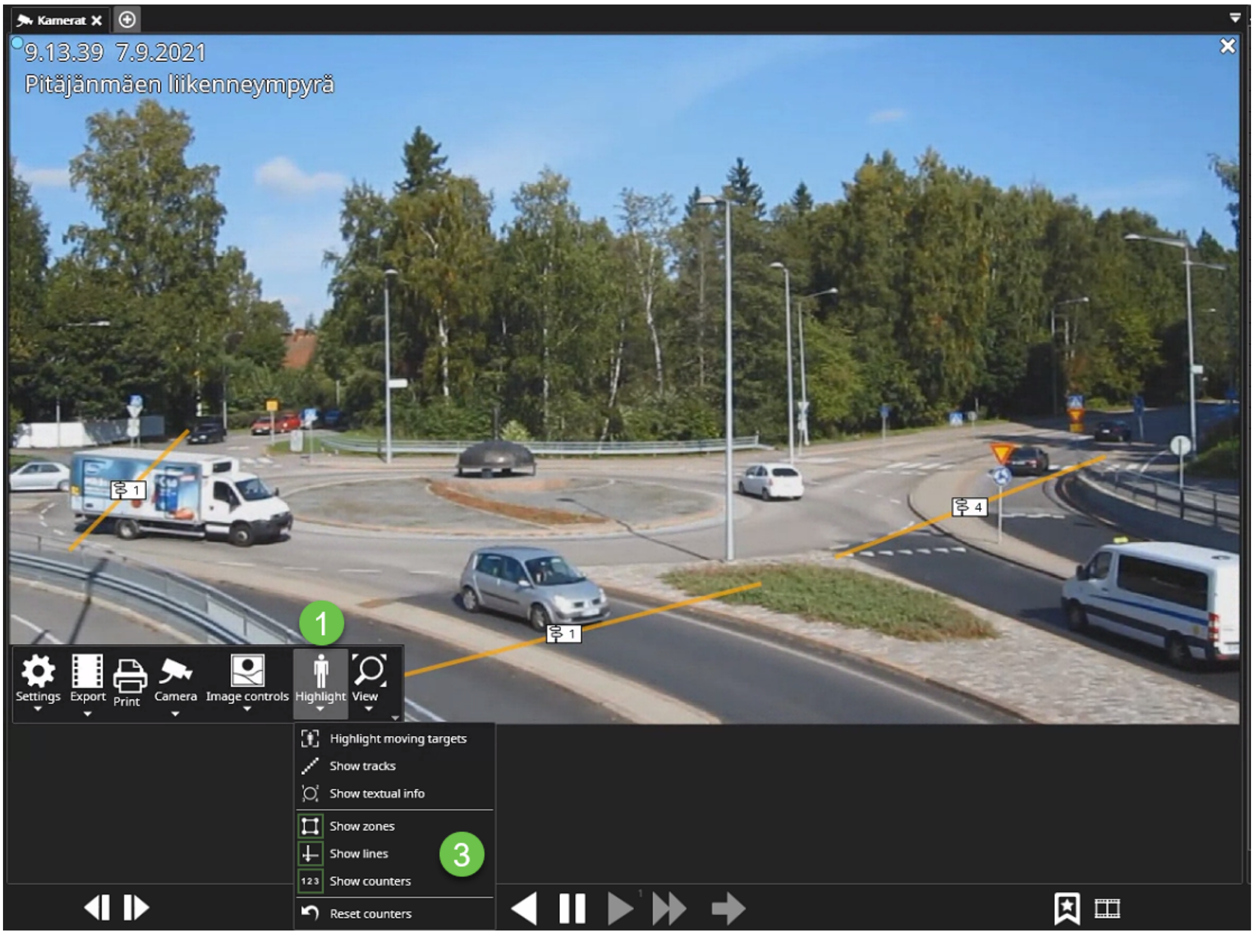
Add Classifier +

⋮	Name:	Person	Speed:	0	to	20	km/h	Area:	0.5	to	2	m^2	🗑
⋮	Name:	Vehicle	Speed:	0	to	200	km/h	Area:	4	to	100	m^2	🗑
⋮	Name:	Clutter	Speed:	0	to	50	km/h	Area:	0	to	0.4	m^2	🗑
⋮	Name:	Group Of People	Speed:	0	to	20	km/h	Area:	2.1	to	3.9	m^2	🗑

[Top](#) [Previous](#) [Next](#)

10. How to view VCA events in the Mirasys Spotter

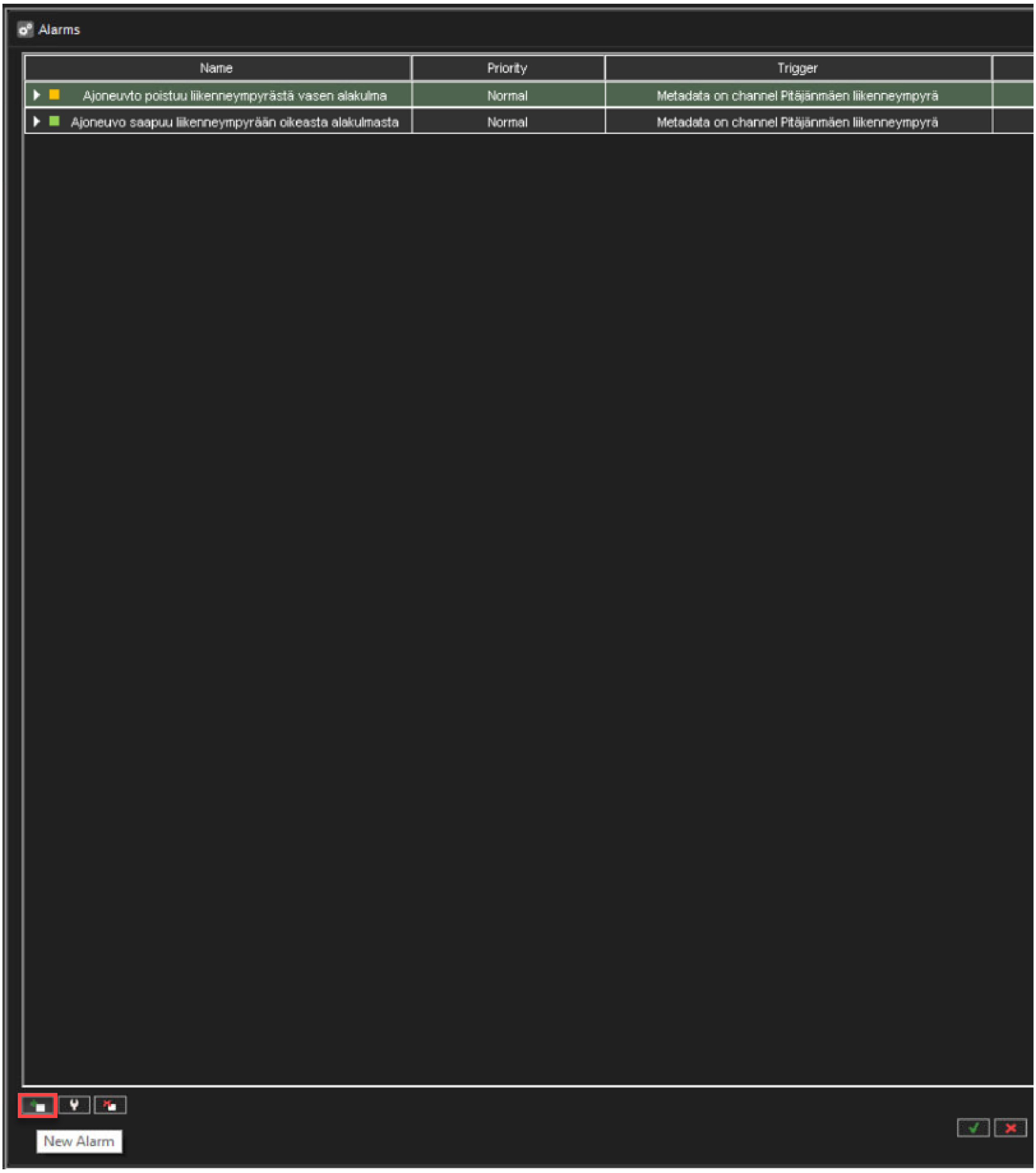
1. Open camera to the real-time view
2. Open camera toolbar and click **Highlight**
3. Enable needed options(**Show zone**s, **Show lines** or **Show counters**)



Previous Next

11. How to create an alarm from VCA event

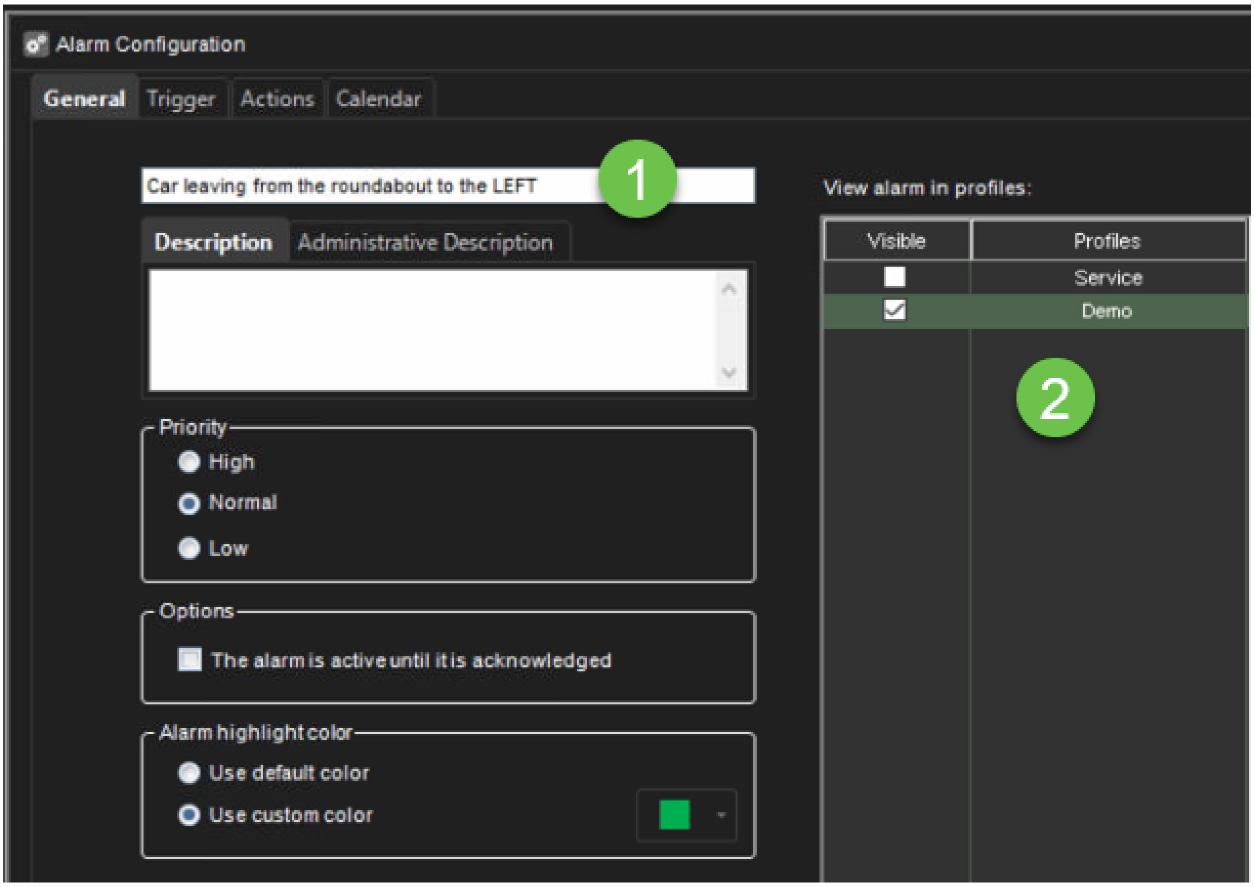
1. Go to the VMS servers tab
2. Click **Alarms**
3. Click **New Alarm**



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

1. Enter the name of the alarm
2. Select the **View alarm in profiles**
3. Go to the **Trigger** tab



Car leaving from the roundabout to the LEFT

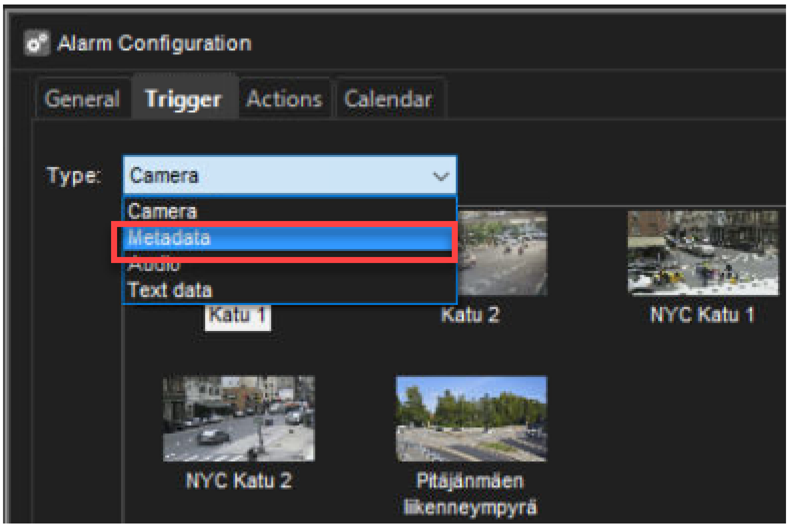
Visible	Profiles
□	Service
☑	Demo

Priority: ○ High ● Normal ○ Low

Options: □ The alarm is active until it is acknowledged

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

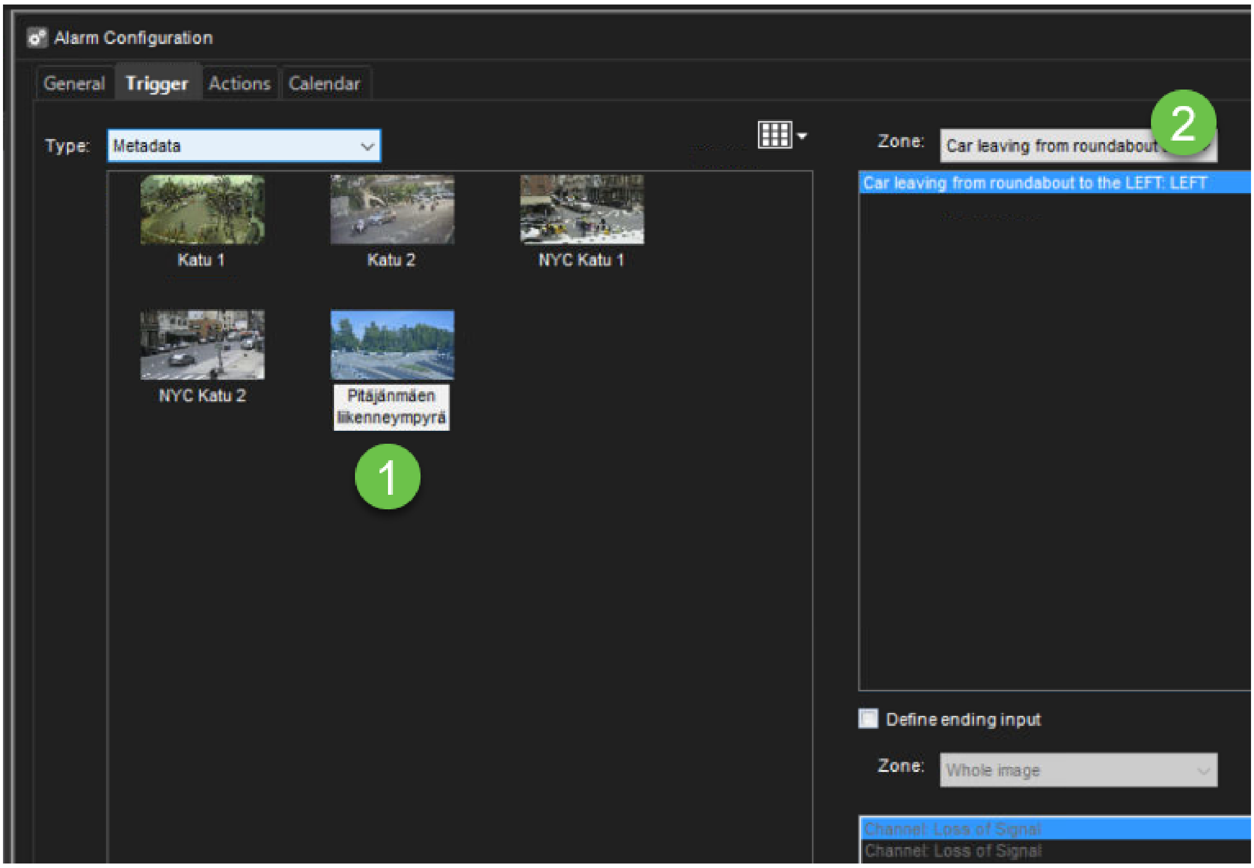
Select **Type**: **Metadata**



Type: Camera
Camera
Metadata
Audio
Text data

1. Select the correct camera from the list
2. Select zone

3. Go to the **Actions** tab



Alarm Configuration

Type: Metadata

Zone: Car leaving from roundabout

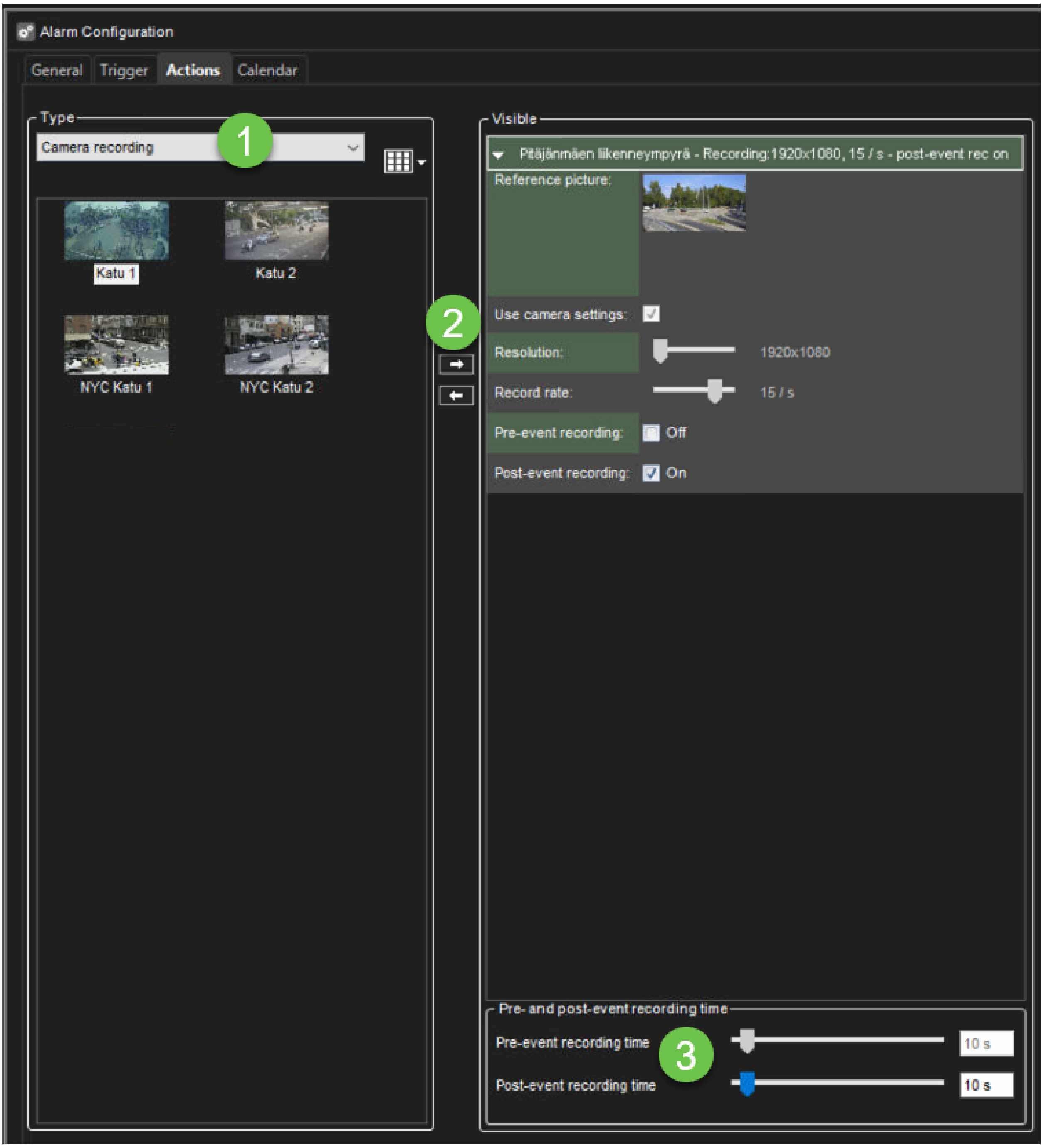
Car leaving from roundabout to the LEFT: LEFT

□ Define ending input

Zone: Whole image

Channel: Loss of Signal
Channel: Loss of Signal

1. Select needed actions
2. Add them to the Visible list
3. Set Pre - and post-event recording time, if needed
4. 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

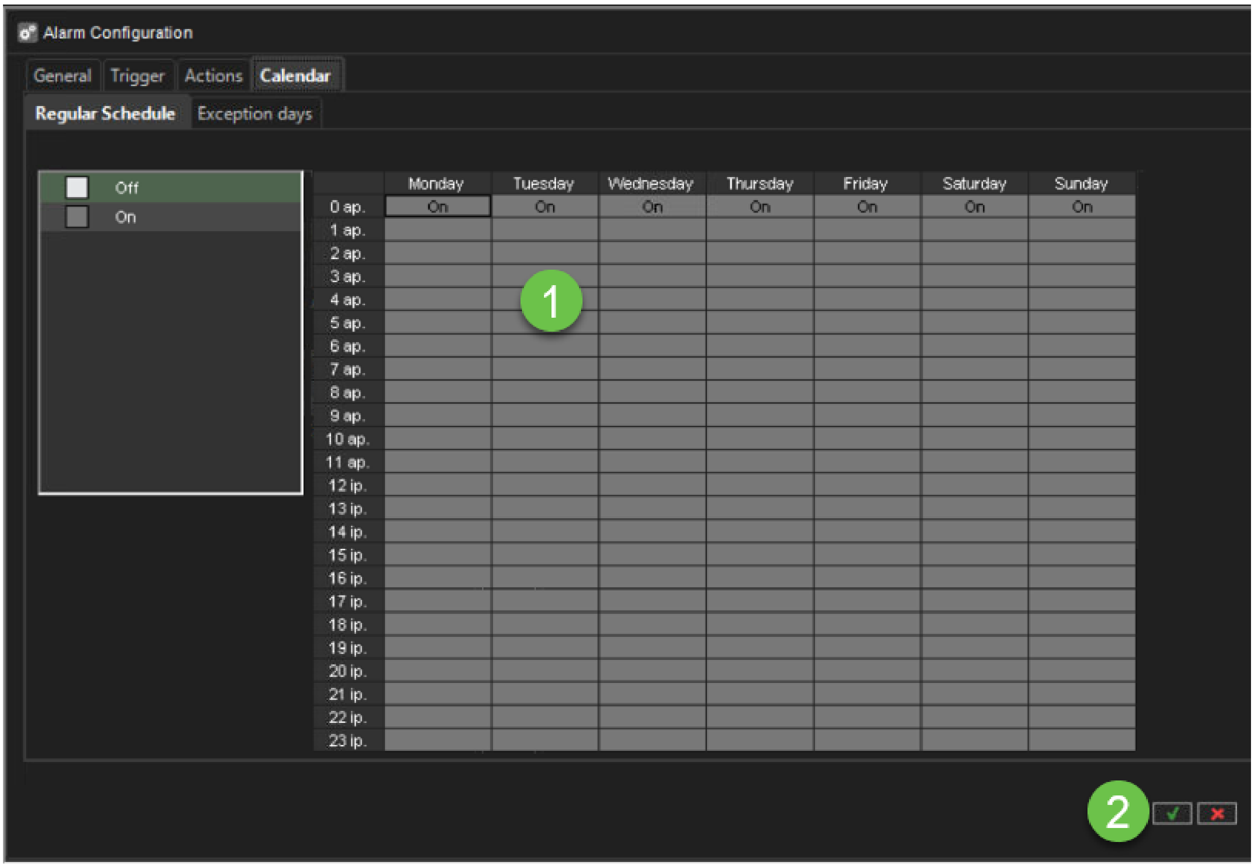
Post-event recording: ☑ On

Pre- and post-event recording time

Pre-event recording time: 10 s

Post-event recording time: 10 s

1. Set alarm schedule
2. Click OK



Alarm Configuration

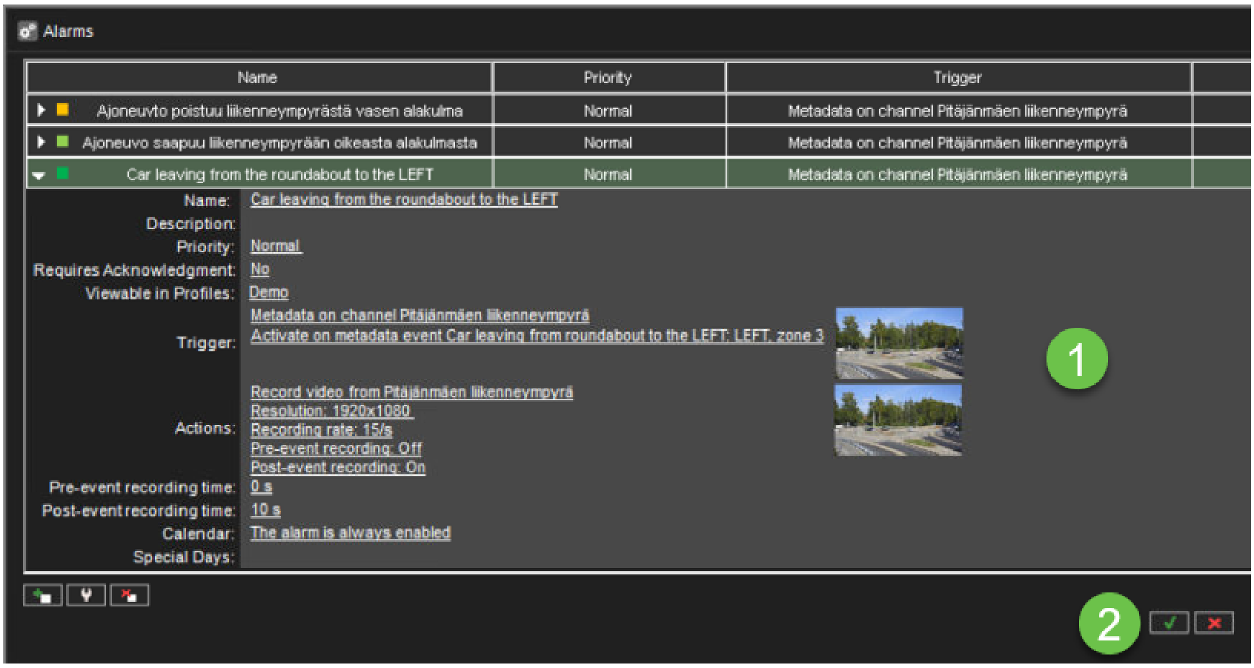
General | Trigger | Actions | **Calendar**

Regular Schedule | Exception days

□ 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.							

1. Check alarm configuration
2. 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:

[Previous](#) [Next](#)

12. Mirasys VCA License Server

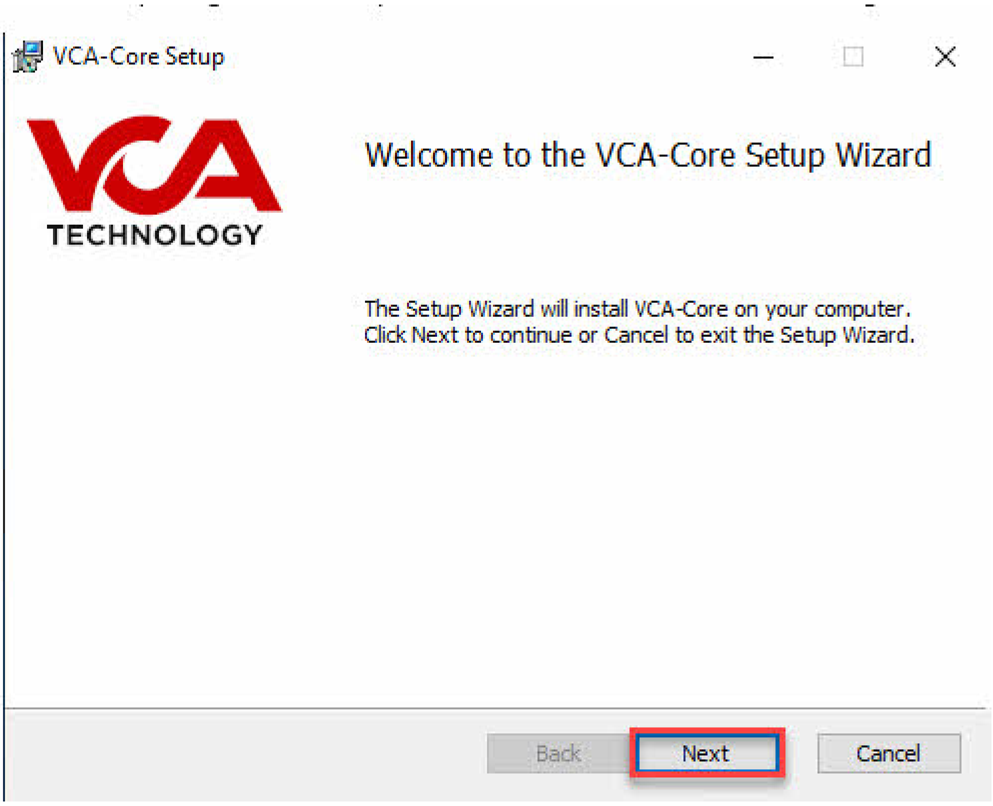
This license server allows the use of VCA in virtual machine/s. For this, you need to install Mirasys VCA License Server to physical hardware and license it.
This server can then share licenses to virtual machine/s.
This feature is supported by 9.4 forward.

Port

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

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

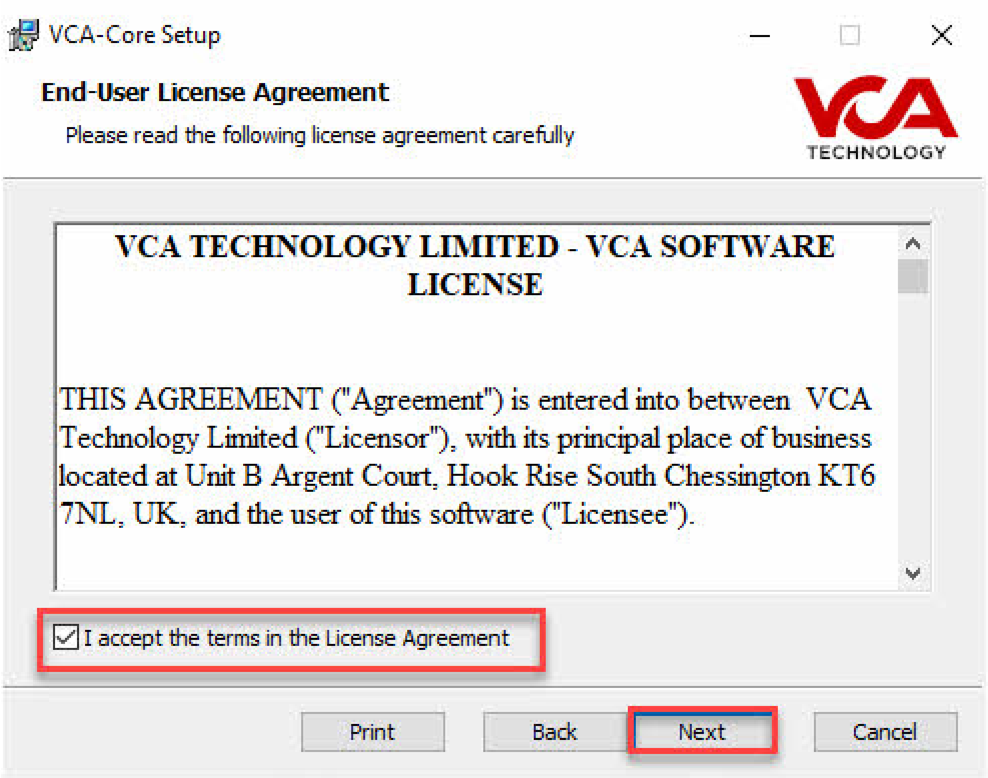


VCA-Core Setup

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

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").

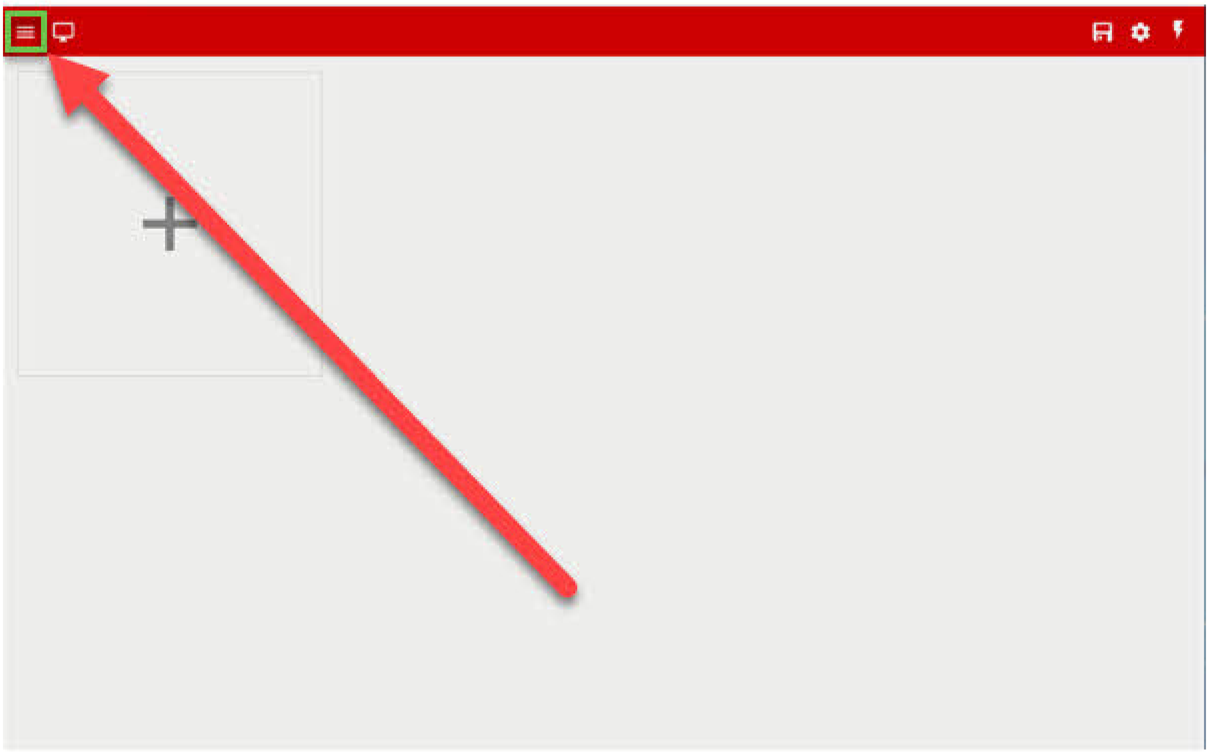
☑ I accept the terms in the License Agreement

5. Follow instructions until the installation is finalized

Usage and licensing

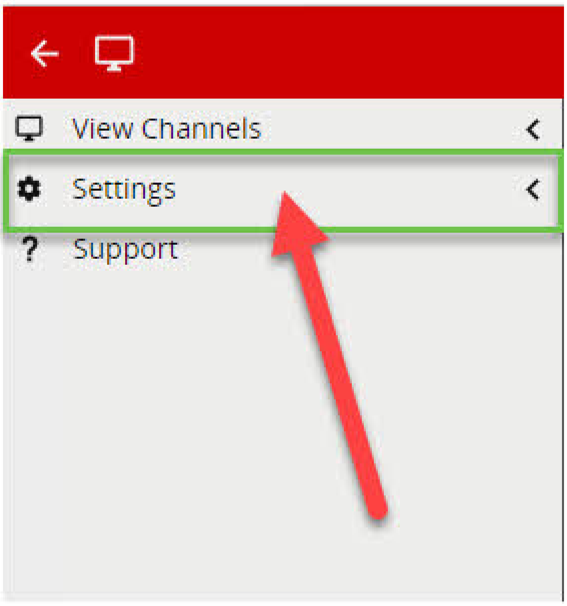
To log in to Mirasys VCA License Server, you need to use the browser and go to the site http://localhost:8080/.
The default username is **admin,** and the default password is **admin**

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

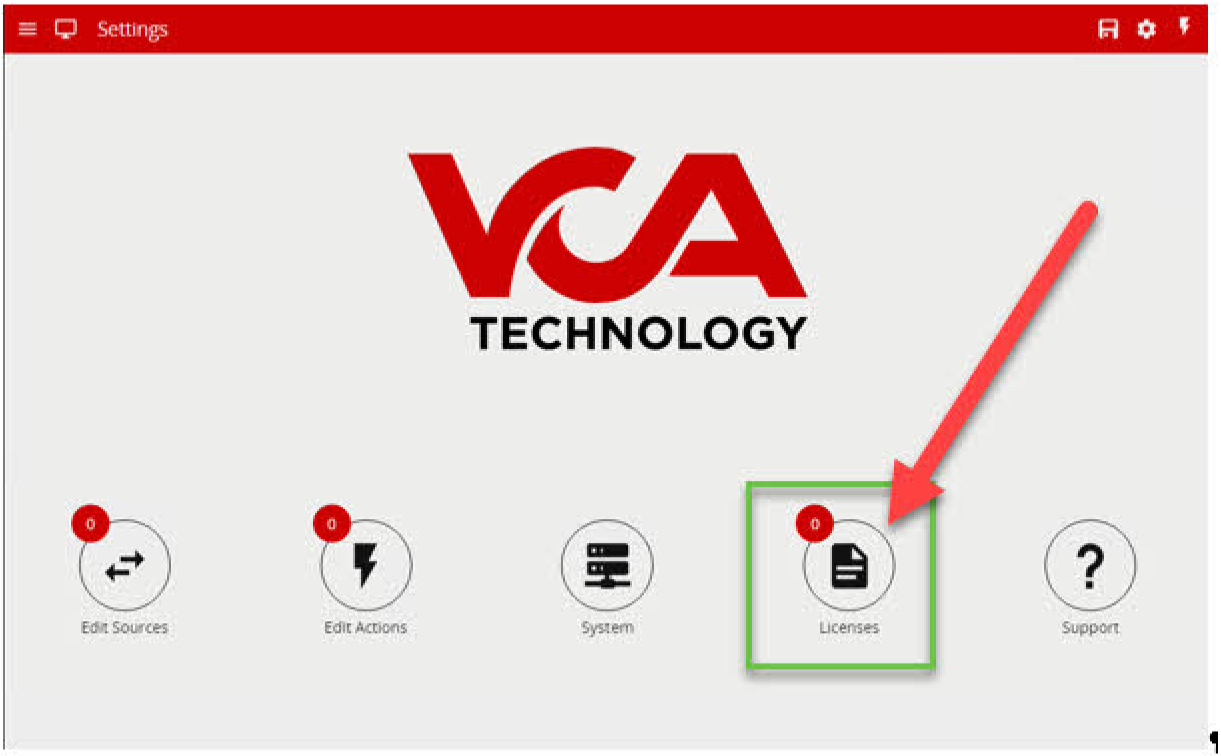


Adding the license

1. Open **Settings**

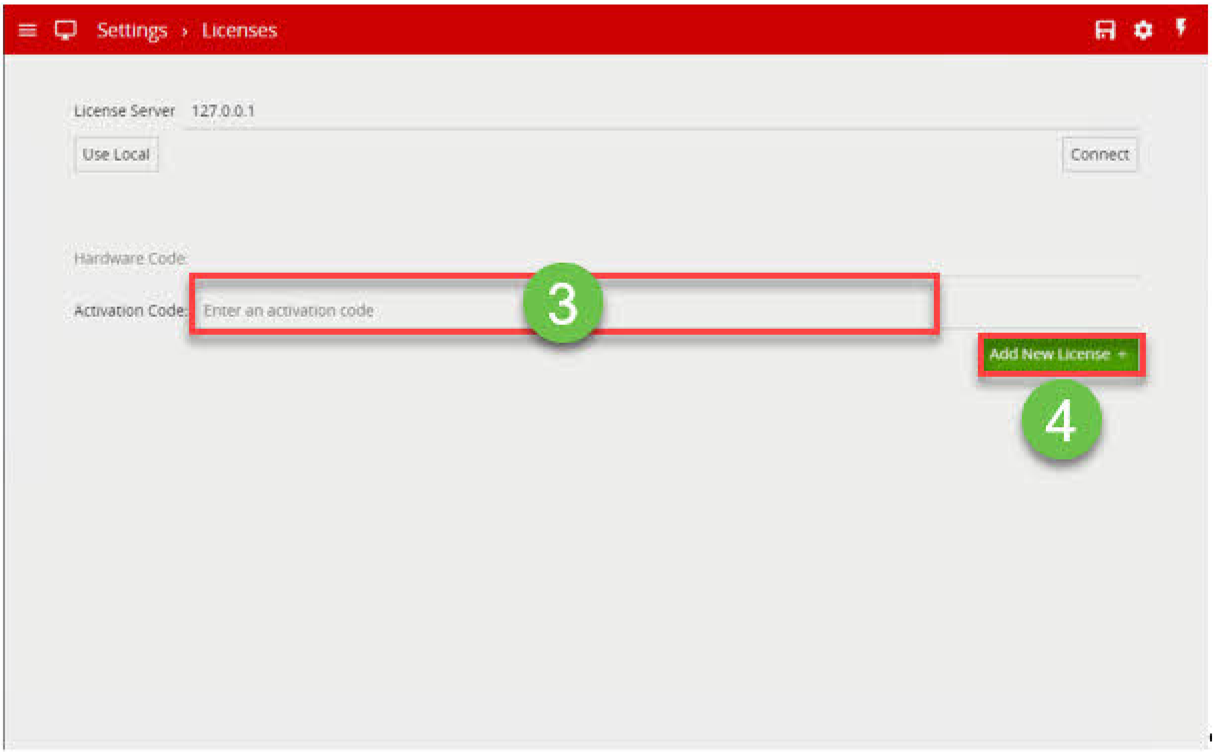


2. Open **Licenses**



Copy Hardware Code and send it to Mirasys(**orders@mirasys.com**)

3. When you have received the activation code from Mirasys, paste the code to the **Activation Code** field
4. And click **Add the new license**



License Server 127.0.0.1

Use Local

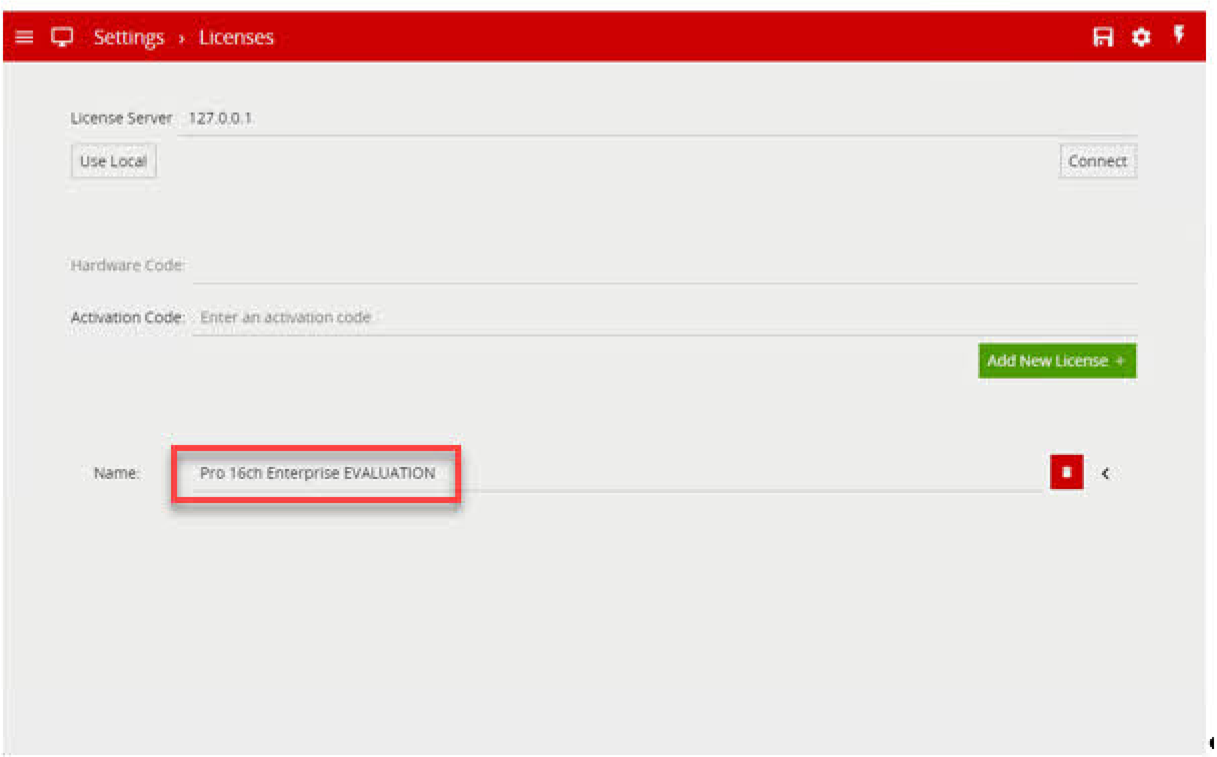
Connect

Hardware Code:

Activation Code: Enter an activation code

Add New License +

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



License Server: 127.0.0.1

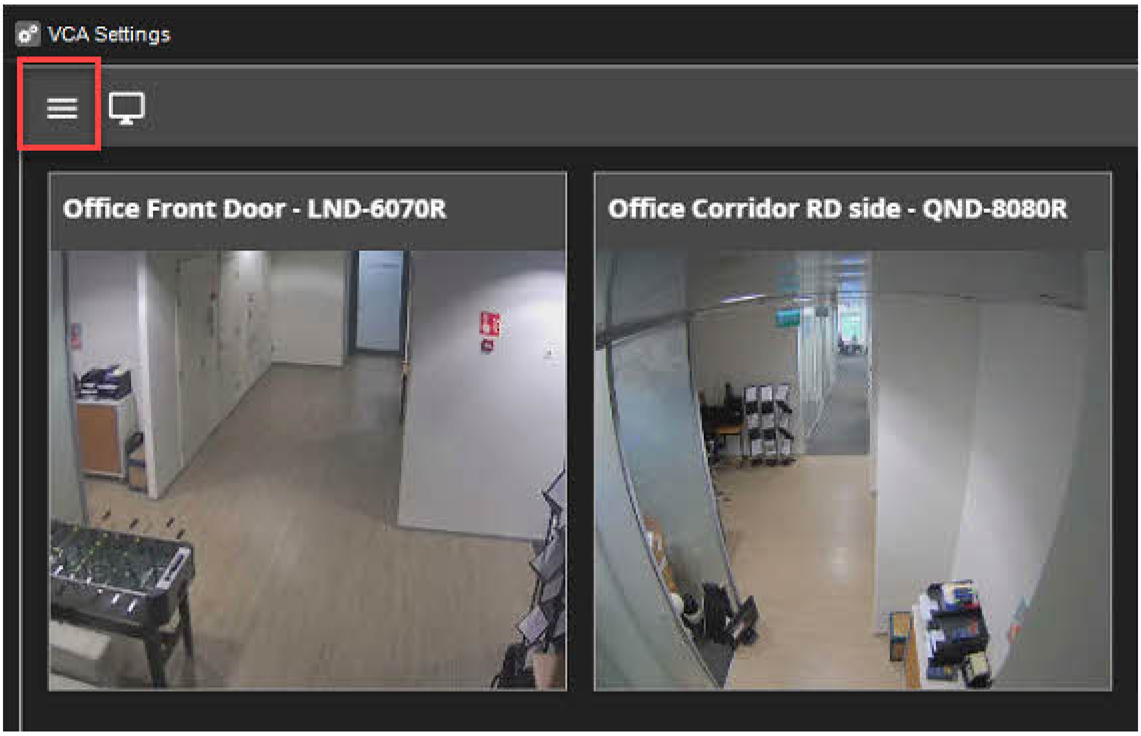
Hardware Code:

Activation Code: Enter an activation code

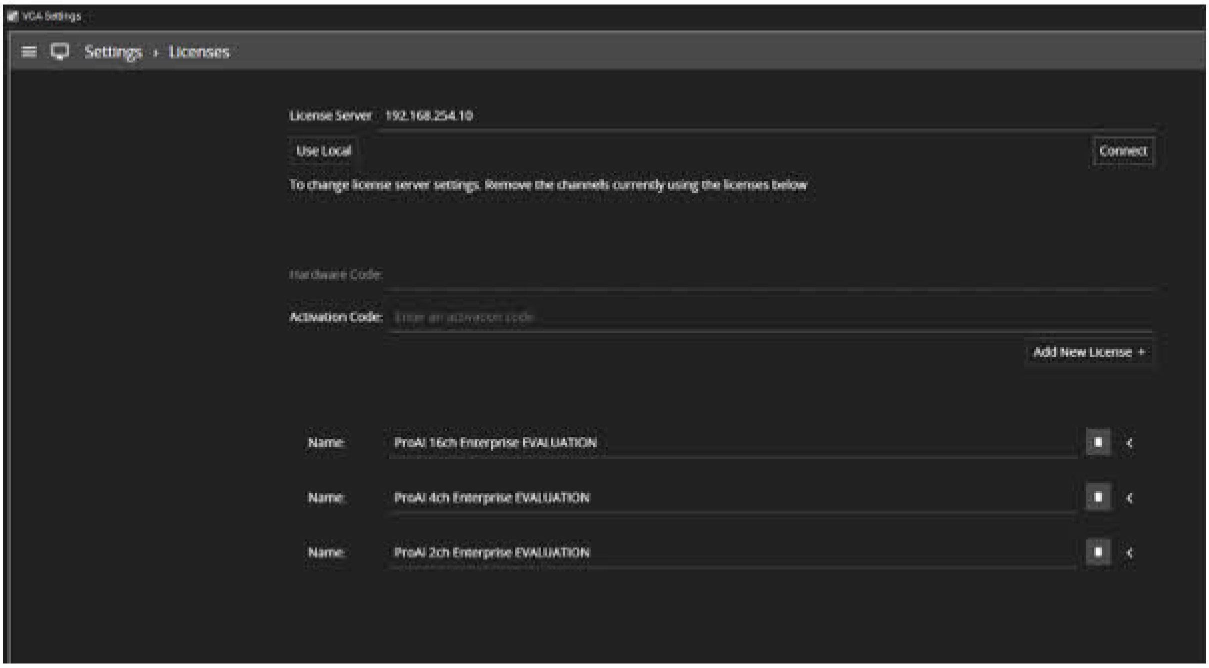
Name: Pro 16ch Enterprise EVALUATION

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

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

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.

Previous Next

13. Mirasys VCA Deep Learning

Requirements

- Nvidia GPU with CUDA cores
    - Pascal or newer
    - Depending on GPU CUDA cores, how many Deep Learning channels you can use on the system
- Latest Nvidia drivers
- CUDA Toolkit 11.1 Update 1
- Mirasys VMS 9.4 or newer
- Deep Learning object files

Installation

1. Install latest Nvidia drivers to the system
2. Download **Mirasys VCA-Deep-Learning-V1-5-8** package from Mirasys Extranet
3. Extract the package
4. Install **CUDA Toolkit 11.1** Update 1 with all features
    a. Some features are not installed because Microsoft Visual Studio is not needed to install but the toolkit is providing example files
    b. If you have installed already Mirasys VMS, before copying files VMS services need to stop
5. Stop services: **WDServer**, **DVRServer** and **SMServer**
6. Copy VCA Deep Learning files from VCA Deep Learning folder to **C:\Program Files\DVMS\DVR\vca\bin** location

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

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

- People Tracking
- Object Tracking
    - Person
    - Cyclist
    - Car
    - Van

- Truck
- Motorcycle
- Bicycle
- Bag

Licensing is done via local VCA Deep Learning licensing or using License Server

Previous