SAFR® Inside

Documentation Version = 3.000

Publish Date = October 21, 2020

Copyright © 2020 RealNetworks, Inc. All rights reserved.

SAFR® is a trademark of RealNetworks, Inc. Patents pending.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Contents

1 Overview

SAFR is an exceptionally accurate AI-powered facial recognition system that provides a high level of visibility and situational awareness for security professionals. You can easily integrate access control peripherals such as cameras, door locks, or alert systems in order to manage access to a location based on people's identities. SAFR runs on a variety of operating systems, including Windows, macOS, Linux, iOS, and Android.

SAFR Inside is an embedded version of SAFR's Video Recognition Gateway (VIRGO) service which performs video analysis and face detection directly inside supported cameras. Simply connect an AXIS Q1615 Mk III camera that has SAFR Inside installed to a SAFR Server (whether deployed on-premise or to the cloud) to perform real-time face recognition. SAFR Inside gives you the ability to leverage powerful computer vision features with dramatic reductions in Total Cost of Ownership (TCO) compared to traditional deployments. By performing key pieces of the image processing (namely face detection and image cropping) directly on the AXIS camera, bandwidth use and other associated costs are dramatically reduced, thus allowing you to deploy facial recognition with less powerful and/or fewer servers.

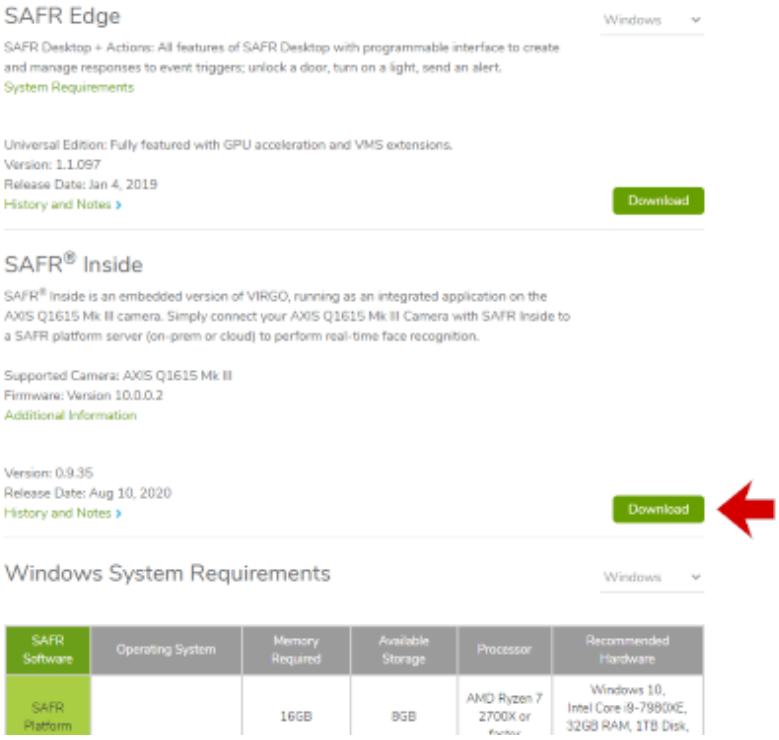
Specifically, SAFR Inside's video feeds are able to do the following:

- Detect faces
- Detect when people are wearing masks
- Identify a person's age
- Identify a person's gender
- Identify a person's sentiment (i.e. happy or unhappy)
- Learn and recognize faces whether or not they're wearing a mask
- Learn and recognize faces that are partially occluded by sunglasses, ballcap, scarves, etc.

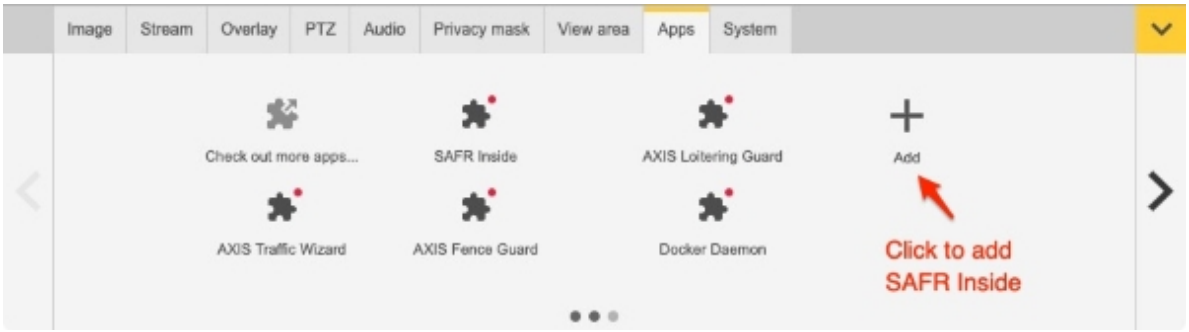
2 SAFR Inside Installation

To install SAFR Inside, do the following:

1. Download the SAFR Inside eap file from the SAFR Download Portal.



2. Log in to your Axis camera by going to its URL and entering your Axis camera credentials.
3. Click on `Settings` in the bottom right corner.
4. Select the `Apps` tab.
5. Click on the `+` icon.



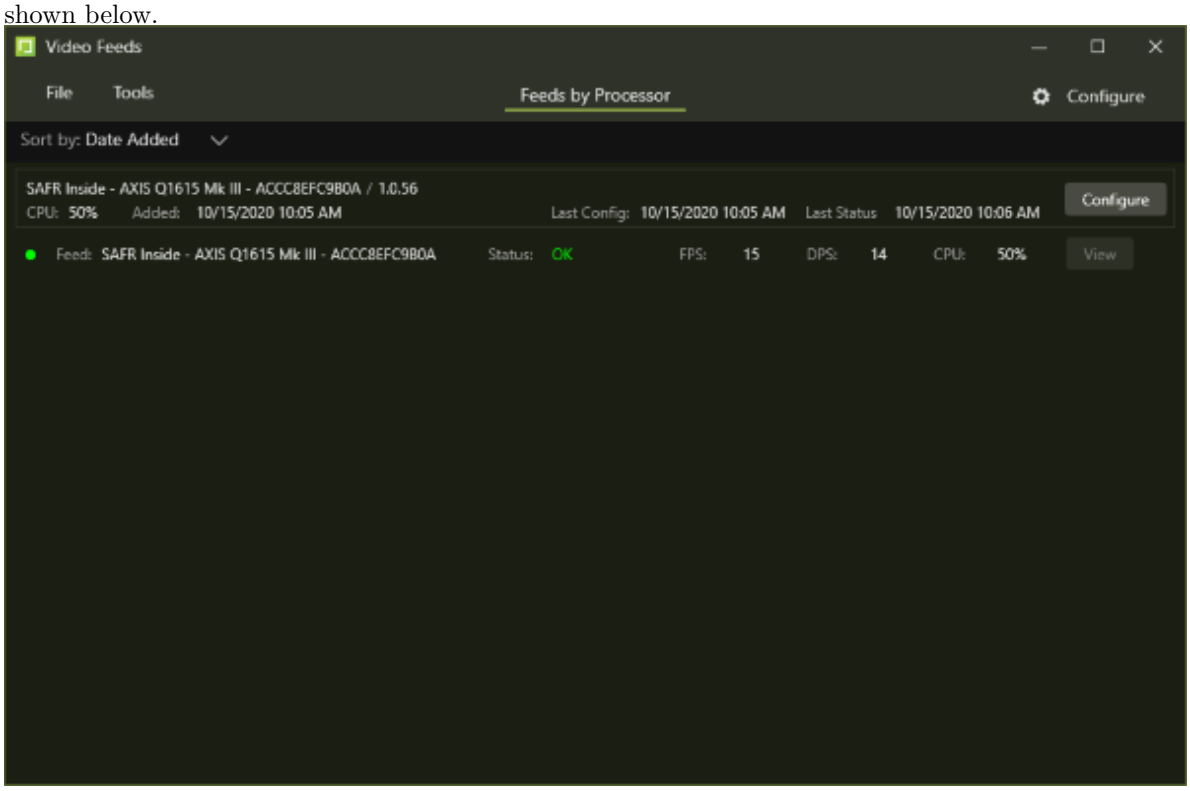
6. In the dialogue, browse to the eap file that you downloaded in step 1, and select `Install`.

SAFR Inside is now installed on your Axis camera.

2.1 Check Installation

To verify that your SAFR Inside installation is working, do the following:

1. Open either the Video Feeds Window of the Desktop client or the Video Feeds Page of the Web Console.
2. Look for an entry for your camera. It's name will probably contain the word "Axis" in its name.
3. Ensure that there is an active feed associated with your camera's entry. Its status should be *OK*, as shown below.

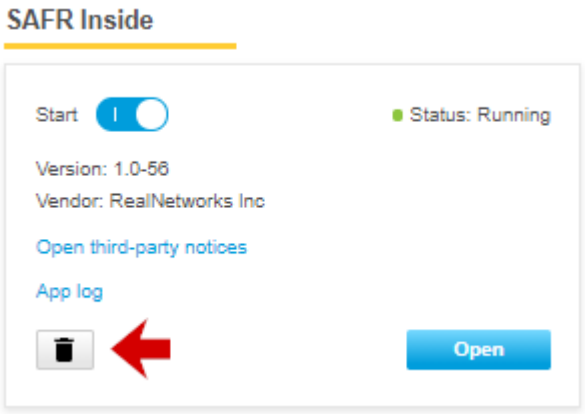


2.2 Uninstall

To uninstall SAFR Inside from the Axis camera, do the following:

1. Go to **Settings -> Apps** on the Axis camera command console.
2. Double click on **SAFR Inside**.

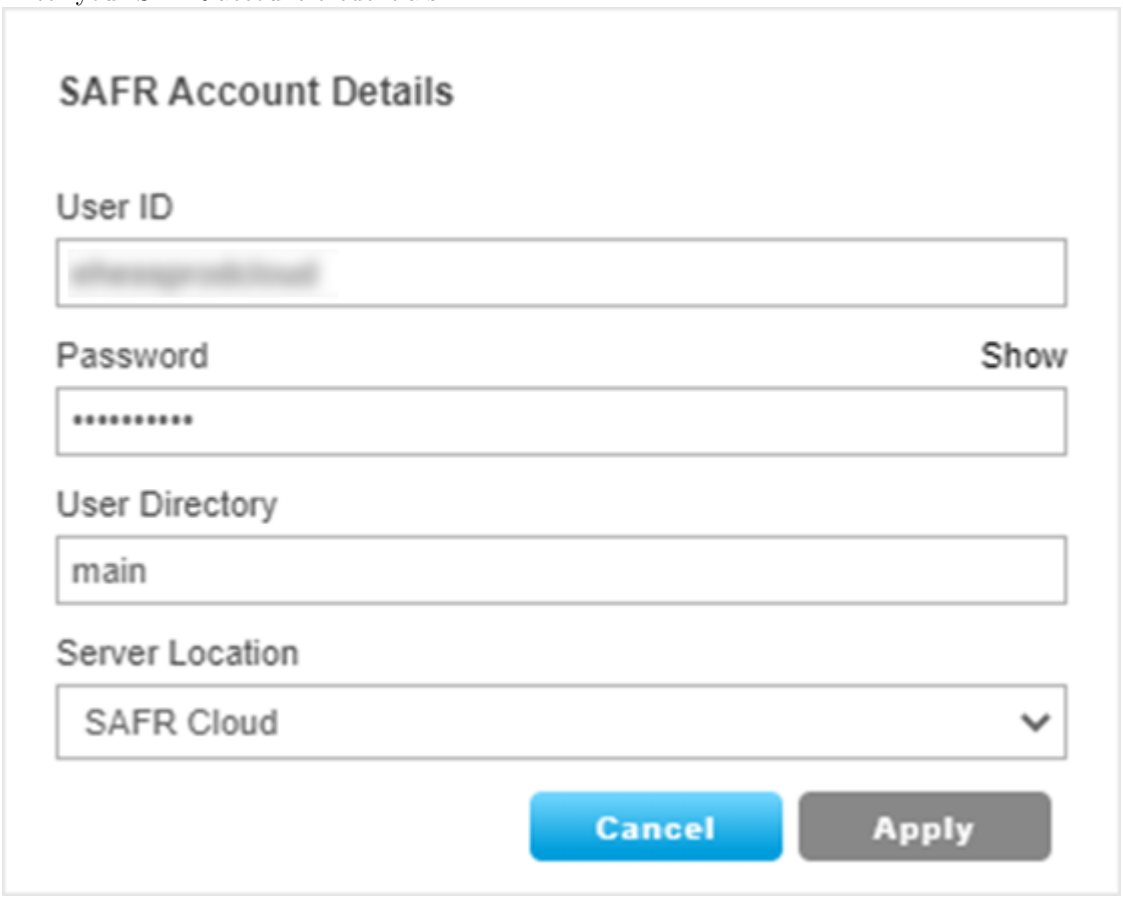
3. Click on the garbage can icon to uninstall SAFR Inside.

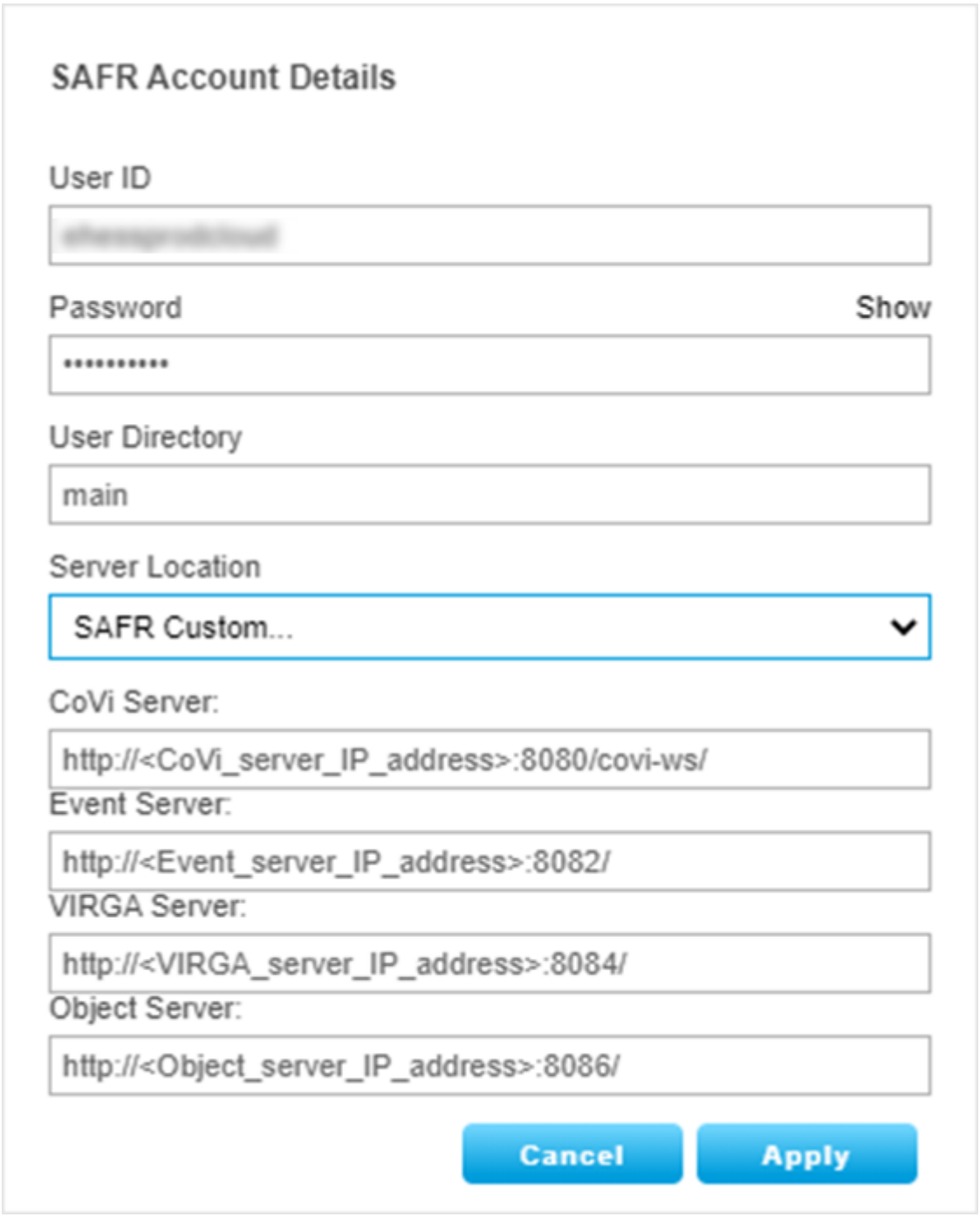


3 Use SAFR Inside

To start using SAFR Inside, do the following:

1. Navigate to the Axis Apps tab and click on `SAFR Inside`.
2. Click on `Open` in the dialogue.
3. Click on the gear icon under Profile properties. 
4. Enter your SAFR account credentials.





3.1 Overlays

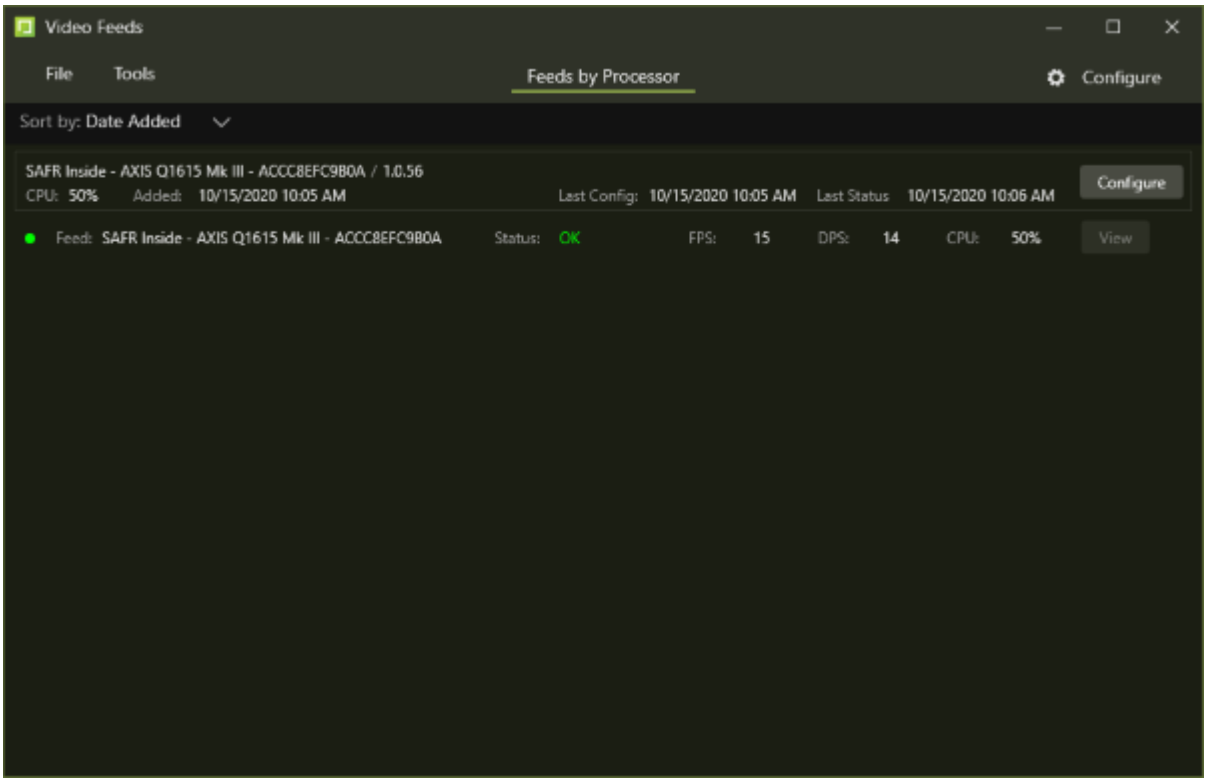
Overlays in SAFR Inside are different than overlays in the Desktop client; in SAFR Inside overlays are intended to be used for setup confirmation and troubleshooting. Overlays shouldn't be on during normal operations because they use the Axis camera's limited CPU processing power, which consequently restricts the maximum number of frames SAFR Inside can process.

- **Display Overlay Information**: Displays the SAFR Inside name, a bounding box for detected faces, and a text indicator if the person is identified (i.e. recognized).
- **Display Identity Information**: Displays the enrolled person's name.

Overlays are visible in the SAFR Inside Settings page only, and do not appear in the camera's *Live View* pages or RTSP video streams.

4 Manage Video Feeds

To manage video feeds, open the *Video Feeds Window* within either the Desktop client or the Web Console, then click the **Configure** button. You'll see a screen similar to the following: (**Note**: You can often expose additional properties by clicking on the arrow next to entries, as shown by the arrow next to the *SomeFeed* entry below.)

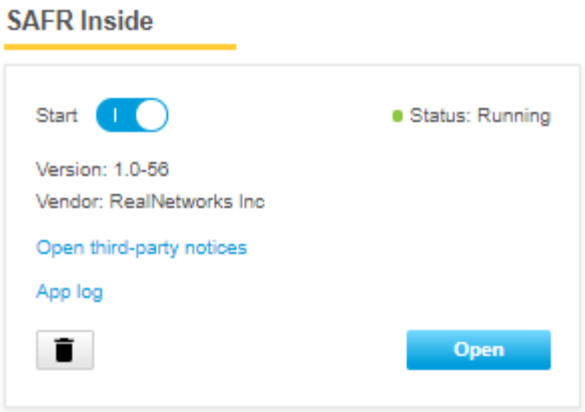


- **name**: The Bonjour/Universal Plug and Play (UPnP) name of the camera on which SAFR Inside is running.
- **admin**: These properties aren't supported for SAFR Inside.
- **global**: Contains global properties. Only one of the global properties is supported for SAFR Inside:
  - *status-interval*: The status reporting time interval in milliseconds.
- **monitoring**: Monitoring properties aren't supported for SAFR Inside.
- **feeds**: Specifies the feed properties. See Appendix: Video Feeds Properties for information about all the video feeds supported for SAFR Inside.
- **update**: Update properties aren't supported for SAFR Inside.

4.1 Terminate a Video Feed

SAFR Inside video feeds that haven't been terminated will continuously run in the background, and will automatically restart themselves after camera shutdowns and restarts. To terminate video feeds that are no longer of interest to you, do the following:

1. Navigate to the SAFR Inside application page in the desired AXIS camera.
2. Click on the start toggle button to terminate SAFR Inside.


SAFR Inside
Start
Status: Running
Version: 1.0-56
Vendor: RealNetworks Inc
Open third-party notices
App log
Open

5 Manage Person Records

Learned identities are saved as person records, which are stored in the Identity Database. The Identity Database can be managed via the People Window of the Desktop client or via the People Page of the Web Console.

5.1 Identity Attributes

Some of the identity attributes are exposed in the default view of the Person Window, but if you double click on people's faces, you can view and configure all the identity attributes. Note that most of their identity attributes will be empty until the information is manually entered.

- **Identifier**: The person's unique identifier within SAFR. This value is automatically assigned to them when they're registered, and cannot be changed.
- **Name**: The person's name. Note that if you enter the person's first and last name into this field, SAFR automatically parses the full name and fills out the first and last name fields for you.
- **First Name**: The person's first name.
- **Last Name**: The person's last name.
- **Id Class**: The person's threat level. (i.e. Threat, Concern, or No-Concern)
- **Person Type**: The person's Person Type. Person Types are groupings that you define to differentiate the people registered in your Identity Database. (e.g. "student", "teacher", and "staff")
- **Gender**: The person's gender.
- **DateofBirth**: The person's date of birth.
- **Moniker**: Used to realize two-factor authentication with visual badges.
- **External Id**: If the person has been imported from another database, this value can be used to track the identity in both databases.
- **Company**: The company the person works for.
- **Home location**: The person's Home Location. Home Locations, much like Person Types, are labels that you define to help differentiate the person records registered in your Identity Database.
- **Phone**: The person's phone number.
- **Email**: The person's email address.
- **Tags**: Any custom tags that you have defined. Each person record can have multiple tags assigned to it.
- **Enrollment date**: The date when the person was registered.
- **Enrollment expiration**: The expiration date of the person's enrollment.
- **Enrolled site**: The site where the person was enrolled.
- **Enrolled source**: The camera that enrolled the person.
- **Last modified**: The time when this person record was last modified. (e.g. an attribute was updated, a more recent reference image was uploaded, etc.)
- **Modified by**: The user that made the last modification to this person record.
- **Modified site**: The site from which this person record was last modified.

5.2 Person Record Sorting and Filtering

You can sort the person records based on either their enrollment date or by the people's names.

Similarly, you can filter which person records are visible based on any of the following criteria:

- **Name**: Filter based on the person records' names.
- **Person Type**: Filter based on the person records' Person Types.
- **Id Class**: Filter based on the person records' Id Classes (i.e. their threat levels).
- **Home Location**: Filter based on the person records' Home Locations.

6 View Events

Events are generated when people passing in front of a connected camera meet any number of configurable event-generating criteria. (e.g. the person is a known threat, the person is wearing a mask, somebody enrolls at a registration kiosk, etc.)

Generated events are stored in the Event Archive, and can be viewed from the Events Window of either the Desktop client or the Web Console. Stored events can be sorted by the following criteria:

- **Chronological**: Sort the events based on when they were recorded.
- **Duration**: Sort the events based on how long they last.
- **Name**: Sort the events based on the name of the person that triggered the event, if known.

6.1 Event Filters

You also have the option of filtering what events you want to browse based on any of the following criteria:

- **Date**: The date when the event was recorded.
- **Id Class**: The threat level of the person that triggered the event.
- **Sites**: The camera or set of cameras that recorded the event. **Note**: Usually *Sites* are set to multiple cameras.
- **Sources**: The camera or set of cameras that recorded the event. **Note**: Usually *Sources* are set to single cameras.
- **Name**: The name of the person that triggered the event.
- **Person Type**: The Person Type of the person that triggered the event.
- **Gender**: The gender of the person that triggered the event.
- **Tenure**: The date when the person that triggered the event was registered to the Identity Database.
- **Shortest Gap**: If a person is viewed by one or more cameras multiple times within this time period, all those appearances are considered the same event.
- **Shortest Duration**: The minimum event duration, in milliseconds.
- **Disparate Sources**: If a person is viewed by multiple cameras at the same time, all those appearances are considered the same event when this filter is enabled. (Only available in the Desktop client)
- **Mask**: Whether or not the person triggering the event was wearing a mask.

7 Configure SAFR Actions

In SAFR an action is essentially a script/macro that communicates a desired action in a language/protocol the receiving device or system understands. It can be written in any language supported by the computer where Actions Relay Event Service (ARES) is installed. It only needs to be invocable as an executable directly or through the use of another executable (usually a script interpreter such as Python).

7.1 Actions Components

These are the principle components involved with actions:

- **Actions Relay Event Service (ARES)**: ARES is a cross-platform Java application that acts as an event listener that dispatches configured actions in response to events, as defined in the SAFRActions.config file. ARES can provide replies on any event to be handled by the client originating the event and is normally installed as a service by either the SAFR Platform or SAFR Edge installers. It is constantly active and is automatically started by the operating system on power-up.
- **SAFRActions.config**: The SAFRActions.config file defines which events will trigger specified actions. It also can specify additional condition constraints before the action(s) will trigger.
- **SAFR Actions**: Only available on macOS and Windows. SAFR Actions is a GUI tool that makes editing the SAFRActions.config file much easier. It presents the JSON information of the config file in a visual and easy to understand manner and offers drop-down menus so you can quickly and easily see what values are available and valid. SAFR Actions makes the JSON element hierarchies easy to understand, and ensures that your changes will validate against the SAFRActions.config JSON schema.

7.2 SAFRActions.config Overview

Below is a summary of the schema used to define actions. See the SAFR Actions documentation for more detailed information of actions and actions' schemas.

```
<name: value connection attributes>
rules: [
  {
    event: {  },
    triggers: [
        <time of day and week properties>
        actions: [  ],
        reply: {  },
        conditionalReply: {  },
    ],
    excludeDates: [  ]
  }
]
noTriggerReply: {  }
emailDef: [  {  }, {  }, ... ]
smsDef: [  {  }, {  }, ... ]
```

- **rules**:
  - 1 or more rules can be defined.
  - When an event occurs each rule is checked to see if any of its events match.
  - A rule's event matches an occurring event when:
    - All attributes rules[i].events match the event.
  - Each rule has 1 or more triggers.
    - Each trigger inside a matching rule is fired as long as the time of day conditions match.
  - Each trigger has one or more actions.
    - Actions are either:
      - A shell command or a batch/shell script to be executed.

        - A send email command that has the syntax of: `@emailSend <value of emailDef.label>`
      - All actions are run asynchronously unless a *conditionalReply* is specified in which case the first rule is run synchronously (and the return code of that rule is used for the conditionalReply) while all other rules are run asynchronously.
- *noTriggerReply* is used to perform a reply if none of the triggers are fired.
- *emailDef* defines one or more email message attributes (subject, from, message, etc). Before any email can be sent, you must first do the following:
  1. Obtain an SMTP server account that you can use to send emails.
  2. Configure SAFR to use your SMTP server using either the Status Page of the Web Console or SAFR Actions on a Windows machine.
- *smsDef* defines one or more Short Message Service (SMS) messages. Before any SMS messages can be sent, you must first do the following:
  1. Set up an AWS Account which is configured for your region.
  2. Configure SAFR to use your AWS account using either the Status Page of the Web Console or SAFR Actions on a Windows machine.

7.3 Examples

Send email when visitor arrives during work hours.

- rules
  - Rule 1
    - event (hasPersonId=false)
    - trigger (day/hours: 8-5, M-F)
      - action: @emailSend visitorEmail
- emailDef
  - label=visitorEmail
  - subject="Visitor Arrived"
  - message="A visitor has arrived at #I - #S."
  - ...

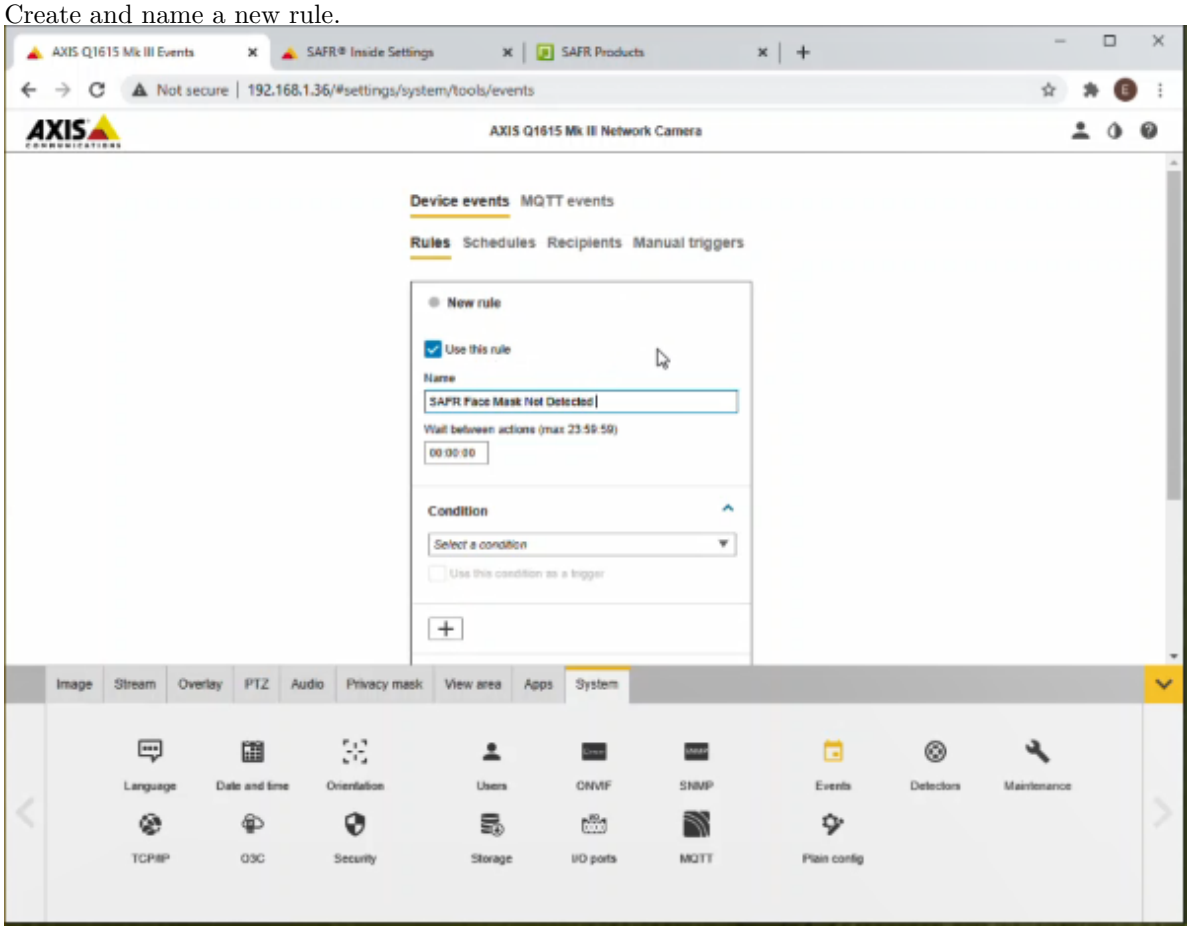
Log all events to a CSV and send one type of email for a known person event and another for a threat event.

- rules
  - Rule1 (known person email)
    - event (hasPersonId=true, idClass=No-Concern)
    - trigger
      - action: @emailSend knownEmail
  - Rule 2 (threat email)
    - event (hasPersonId=true, idClass=[Threat, Concern])
    - trigger
      - action: @emailSend threatEmail
  - Rule 3 (log)
    - trigger
      - action: ".\scripts\log_event.bat "#D" "#N" "#F" ..."
- emailDef
  - 1 (label=knownEmail, subject, message, etc)
  - 2 (label=threatEmail, subject, message, etc)

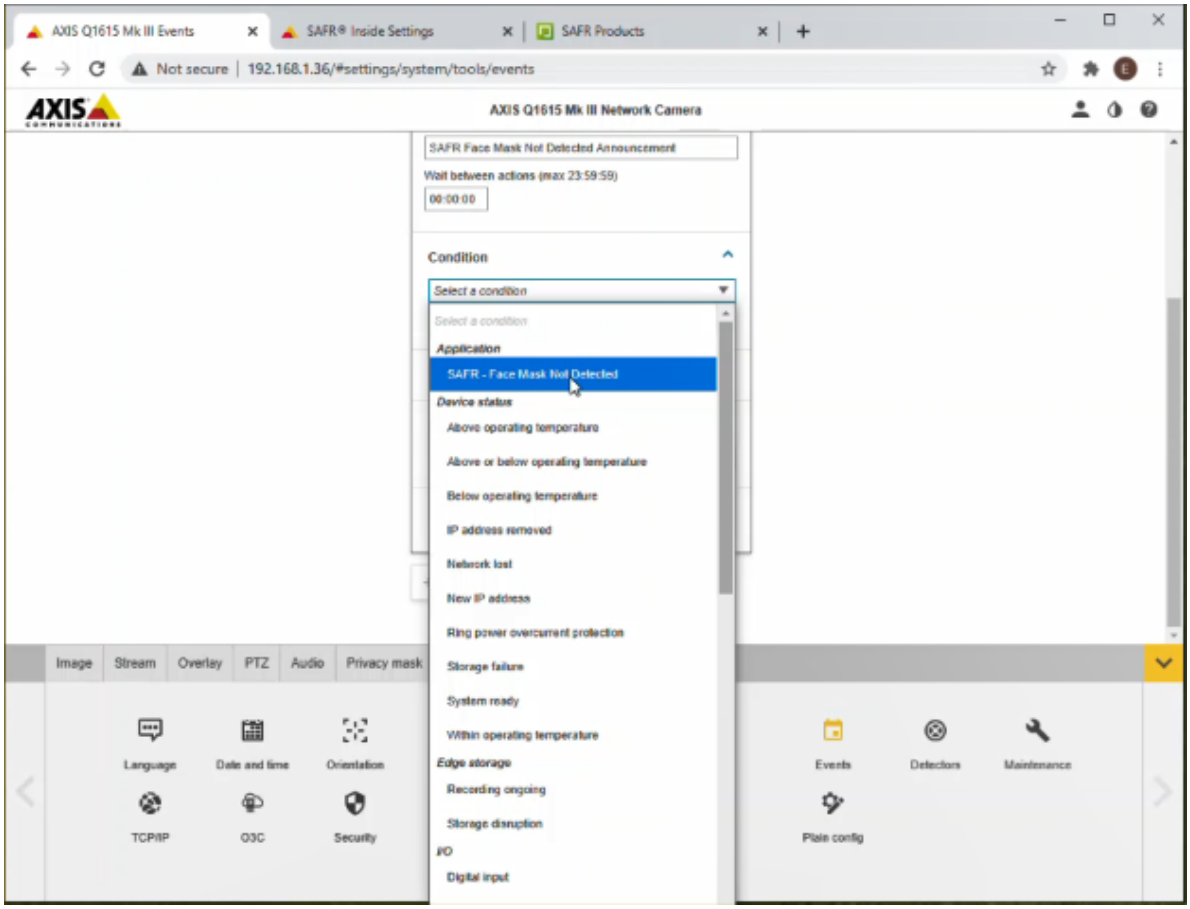
8 Trigger Audio Alerts

You can use SAFR Inside to trigger an audio event using the AXIS camera's *Events and Rules* engine when a face is detected that isn't wearing a mask or PPE.

1. Make sure there are speakers plugged in to the Axis camera; the camera doesn't have any onboard speakers.
2. Open **System -> Events** in the Axis camera app menu.
3. Create and name a new rule.



4. Select the *Application -> SAFR - Face Mask Not Detected* condition from the list of available conditions. **Note**: SAFR Inside must be running for the *SAFR - Face Mask Not Detected* condition to appear in the list of conditions.



5. You may select any audio file that has been uploaded to the camera via the AXIS camera's **Settings -> Audio** tab.
6. If you're using a long audio clip (e.g. more than 10 seconds) you should consider setting the **wait between actions** value in order to avoid having the clip play multiple times on top of itself.

8.1 Sample Audio Alert Sound Files

Here are several sample audio files you could use as audio alerts:

- **Audio alert 1_tone-1.au**: Opening tone 1 followed by "Welcome. Please remember that all guests are required to wear a face mask."
- **Audio alert 1_tone-2.au**: Opening tone 2 followed by "Welcome. Please remember that all guests are required to wear a face mask."
- **Audio alert 2_tone-1.au**: Opening tone 1 followed by "Welcome. Reminder: all shoppers are required to wear face masks. Please see customer service if you need a complementary mask."
- **Audio alert 2_tone-2.au**: Opening tone 2 followed by "Welcome. Reminder: all shoppers are required to wear face masks. Please see customer service if you need a complementary mask."
- **Audio alert 3_tone-1.au**: Opening tone 1 followed by "Welcome. Reminder: face masks are required at all times while indoors. Please see reception if you need a complementary mask."
- **Audio alert 3_tone-2.au**: Opening tone 2 followed by "Welcome. Reminder: face masks are required at all times while indoors. Please see reception if you need a complementary mask."

9 Appendix: Video Feeds Properties

Below are all the feeds properties that can be configured.

Property	Default Value	Description
detector.detect-faces	TRUE	Whether detection of faces should be enabled for this feed.
detector.detect-faces-input-size	“normal”	Sets the face detector input size. This property allows you to manage the trade-off between accuracy vs. speed. There are 3 possible values: **normal** - This is the standard against which the other 2 possible values are measured. **small** - Decreased accuracy but increased speed. **large** - Increased accuracy but decreased speed.
detector.maximum-input-resolution	720	Maximum resolution of the Input image. Bigger images are scaled down (aspect-ratio preserving) to this resolution before detection.
detector.minimum-required-face-size	0	minimum size of faces to accept from the detector. Only faces with at least this size are eligible for recognition.
directory	N/A	Directory name.
enabled	FALSE	Marks the feed as enabled or disabled.
input.back-channel.mobotix.cash-point	“None”	When the connected camera is a Mobotix camera, this property must be set to the configured cash point within the Mobotix app for the back-channel to work.
input.back-channel.type	“None”	When the connected camera is a Mobotix camera, you can set this property to “Mobotix MX” in order to have SAFR report STRANGER and RECOGNIZED event types to the camera. This feature is necessary if you want to make use of the Mobotix app. If the connected camera isn’t a Mobotix camera, this property doesn’t have any effect.
input.stream.url	N/A	The video stream URL. The URL must point to a RTSP, HTTP or FILE stream.
input.type	“stream”	The type of feed input; either “stream” or “file”.

Property	Default Value	Description
mode	"Enrolled and Stranger Monitoring"	Specifies which video processing mode the feed is using.
recognizer.detect-age	FALSE	Enables the detection of age information.
recognizer.detect-gender	FALSE	Enables the detection of gender information.
recognizer.detect-identity	TRUE	Enables detection of an identity, which matches against the existing database of people (identities).
recognizer.detect-mask	FALSE	When enabled, SAFR will evaluate all occluded faces to see if they're covered by a mask. If they are, then SAFR will use the mask enhanced model to attempt to recognize the face behind the mask. If the occluded face isn't covered by a mask, then the normal occluded model will be used instead.
recognizer.detect-mask-model	"precise"	Specifies the model to be used for mask detection. There are 3 possible values: **Precise**: This model produces the least number of false positives (i.e. detecting that a person is wearing a mask but there is no mask), but it suffers from the lowest true positive rate. (i.e. detecting masks that are actually there) **Sensitive**: This model produces the highest true positive rate, but it suffers from the highest number of false positives. **Normal**: This model produces a moderate amount of both false positives and true positives.
recognizer.detect-mask-threshold	0.5	Specifies the threshold at and above which mask detection will conclude that mask=true.
recognizer.detect-occlusion	FALSE	Enables occlusion detection during recognition.
recognizer.detect-sentiment	FALSE	Enables the detection of sentiment information.
recognizer.identity-masked-threshold-offset	0	Sets the identity threshold when detecting masks.
recognizer.identity-proximity-threshold-allowance	0.13	A boost value that is added to the Identity Recognition Threshold.
recognizer.identity-recognition-threshold	0.54	Identity recognition threshold.

Property	Default Value	Description
recognizer.learning-enabled	FALSE	Enables the feed to learn new identities.
recognizer.learn-occluded-faces	FALSE	Enables learning of occluded faces regardless of the maximum occlusion setting. If this is true then the server configuration will be used, which by default doesn't do any occlusion detection.
recognizer.maximum-clip-ratio	0.1	The maximum clip ratio on either side the recognition candidate might have.
recognizer.maximum-clip-ratio-identification	0	The maximum clip ratio on either side the insertion candidate might have.
recognizer.maximum-pitch-identification	0.4	The maximum pitch value used to determine if the face is looking straight ahead. The pitch value is the forward/backward movement of the face.
recognizer.maximum-roll-identification	0.15	The maximum roll value used to determine if the face is looking straight ahead. The roll value is the side to side tilt movement of the face.
recognizer.maximum-yaw-identification	0.4	The maximum yaw value used to determine if the face is looking straight ahead. The yaw value is the side to side movement of the face.
recognizer.max-occlusion	0	The maximum occlusion value that is allowed when adding a new candidate images into the Person Directory. If the face is occluded with a value greater than this then the face will not be added, but if it's less than or equal to this value then it will be added.
recognizer.minimum-center-pose-quality	0.05	The minimum center pose quality that a recognition candidate must have in order to allow the addition of the candidate image into the Person Directory.
recognizer.minimum-center-pose-quality-identification	0.45	The minimum center pose quality that a recognition candidate must have in order to allow the addition of the candidate image into the Person Directory.

Property	Default Value	Description
recognizer.minimum-center-pose-quality-merging	0.59	The minimum center pose quality that a recognition candidate must have in order to allow merging.
recognizer.minimum-face-contrast-quality	0.1	The minimum face contrast quality that a face image must have before recognition is attempted.
recognizer.minimum-face-contrast-quality-identification	0.3	The minimum face contrast quality that a recognition candidate must have in order to allow the addition of the candidate image into the Person Directory.
recognizer.minimum-face-contrast-quality-merging	0.45	The minimum face contrast quality that a recognition candidate must have in order to allow merging.
recognizer.minimum-face-sharpness-quality	0.1	The minimum face sharpness quality that a face image must have before recognition is attempted.
recognizer.minimum-face-sharpness-quality-identification	0.3	The minimum face sharpness quality that a recognition candidate must have in order to allow the addition of the candidate image into the Person Directory.
recognizer.minimum-face-sharpness-quality-merging	0.45	The minimum face sharpness quality that a recognition candidate must have in order to allow merging.
recognizer.minimum-face-size	80	The minimum size of faces to detect. This value is applied after searching the image.
recognizer.minimum-face-size-identification	120	The minimum resolution that a recognition candidate image must have in order to allow the addition of the candidate image into the Person Directory.
recognizer.minimum-face-size-merging	220	The minimum resolution a recognition candidate must have in order to allow merging.

Property	Default Value	Description
recognizer.pose-configuration-identification-enabled	FALSE	If this is true then pose configuration is enabled for identification. The pose configuration allows for replacing center pose quality with advanced parameters such as yaw, pitch and roll. When pose configuration is enabled, then recognizer.minimum-center-pose-quality is ignored and the following 3 properties are used instead: recognizer.maximum-yaw-identification, recognizer.maximum-pitch-identification, and recognizer.maximum-roll-identification.
reporter.delay	0	Delay the event reporting to the server by this amount in seconds.
reporter.enabled	TRUE	Enables or disables event reporting.
reporter.minimum-event-duration-identified	0	The minimum allowed recognized person event duration in seconds. Events below this value will not be reported.
reporter.minimum-event-duration-unidentified	1500	The minimum allowed unrecognized person event duration in seconds. Events below this value will not be reported.
reporter.report-event-face	TRUE	Enables the inclusion of face thumbnails in event reports.
reporter.report-event-scene	FALSE	Enables the inclusion of scene images in event reports.
reporter.report-secondary-events	FALSE	Reports secondary events. Secondary events are events that are associated with a primary event via the rootEventId property in the event. It is usually preferred to only report the primary events and the secondary events need to only be reported if there is more detail needed. If this is disabled then all events with a rootEventId property set to a primary event will not be reported. Only events with rootEventId not set to anything will be reported, which are the primary events.

Property	Default Value	Description
reporter.report-speculated-events	TRUE	Reports events for speculated faces. Speculated faces are faces that aren't a 100% match, but are close.
reporter.report-stranger-events	TRUE	Reports events for people that are strangers. These are people not registered by the system after running facial recognition on the face.
reporter.report-unrecognizable-events	TRUE	Reports events for people that are not recognizable.
reporter.stranger-events.only-if-occluded	FALSE	Specifies whether only occluded stranger events should be reported. By default stranger events are only generated if the face is not occluded, if occlusion detection is enabled, otherwise they are generated when the face meets the identification image quality metrics. If this option is set to true then stranger events will be reported only if the face is occluded.
reporter.stranger-maximum-age	0	The maximum age of strangers that will trigger stranger events. If a stranger older than the specified maximum age is detected, no stranger event is generated.
reporter.stranger-minimum-age	0	The minimum age of strangers that will trigger stranger events. If a stranger younger than the specified minimum age is detected, no stranger event is generated.
reporter.update-images	FALSE	Updates the thumbnail images with higher quality images during the course of the event if possible.
site	N/A	Site name, if any.
source	N/A	Source name.
statistics.enabled	FALSE	Specifies whether VIRGO should record and report statistics for this feed.
tracker.enable-face-bounds-prediction	TRUE	Enables face bounds prediction, which predicts which direction the face is moving to maintain tracking.
tracker.enable-face-size-correlation	TRUE	Enables face correlation of tracked faces, which compares detected faces looking for a change in area.

Property	Default Value	Description
tracker.identity-relearn-interval-days	0	Updates the identity only when the currently saved identity is older than the updated identity.
tracker.identity-update-better-image	FALSE	Updates the identity in the case where the identity currently saved is of lower quality (in all aspects) than the updated identity.
tracker.initial-recognition-attempts	3	The number of initial recognition attempts to make on an unrecognized person as fast as possible.
tracker.maximum-linger-frames	30	Determines for how many frames more we continue to keep a tracked face around after we have failed to detect it in the most recent frame. This makes the tracker resilient against intermittent loss of face.
tracker.max-position-change-relative-to-face	115	The maximum position change, specified in percentage relative to the face size, to continue tracking.
tracker.max-size-change-relative-to-face	50	The maximum size change, specified in percentage relative to the object size, to continue tracking.
tracker.min-failed-recognitions-to-stop-tracking-identity	3	When a face is being tracked recognitions are continually confirming the identity. The identity is also being verified if it is transferred from a person object. In these cases, if the recognition or verification fails this number of consecutive times then the identity will be reset and no longer associated with the face because we are no longer sure it is the same identity.
tracker.minimum-number-identical-recognitions-learn	2	The number of consecutive recognitions that need to occur before adding a new identity to the system.
tracker.minimum-number-identical-recognitions-lock	1	The number of consecutive recognition attempts that we must run and produce the same person identity before we lock onto this identity.
tracker.reconfirmation-interval	1000	Identity reconfirmation time interval in ms.

Property	Default Value	Description
tracker.stop-tracking-on-failed-re-recognition	FALSE	If recognition fails when re-recognizing a person then delete the identity that was created.