SENSTAR™

Senstar Symphony SDK

8.x

Developer Guide

Contents

Introduction

The Senstar Symphony SDK provides sample code that helps you to build applications to extend the functionality of and integrate third-party products with the Senstar Symphony Server.

The Senstar Symphony Client uses these same methods to ensure quality and feature completeness. Sample applications are included for a wide variety of use cases. Reference documentation is provided in `<SDK>/Documentation/readme.html`.

Note: You might need to restart Senstar Symphony services before a Senstar Symphony SDK application works correctly.

Quick start

Running a Senstar Symphony SDK application

1. Create an Senstar Xnet account.
2. Download and install the Senstar Symphony Server and Senstar Symphony Client.
3. Use a Web browser to access the Senstar Symphony Server configuration interface and add a camera to the Senstar Symphony Server.
4. Download and install the Senstar Symphony SDK. If you install the Senstar Symphony SDK on a computer without the Senstar Symphony Server or Senstar Symphony Client, open a command prompt in the Senstar Symphony SDK bin folder (`C:\Program Files (x86)\Senstar \Symphony SDK\bin` by default) as an administrator and run the following command:

   ```
   for %x in (*.dll *.ax *.ocx) do regsvr32 /s %x
   ```

5. Run `LiveStreamTest.exe` in the Senstar Symphony SDK to see live video from the Senstar Symphony Server.

Compiling a Senstar Symphony SDK application

1. Open `SDK.sln` in Microsoft Visual Studio.
2. Right-click **LiveStreamTest** in the Solution Explorer and click **Set as Startup Project**.
3. Press F5.

Requirements and recommendations

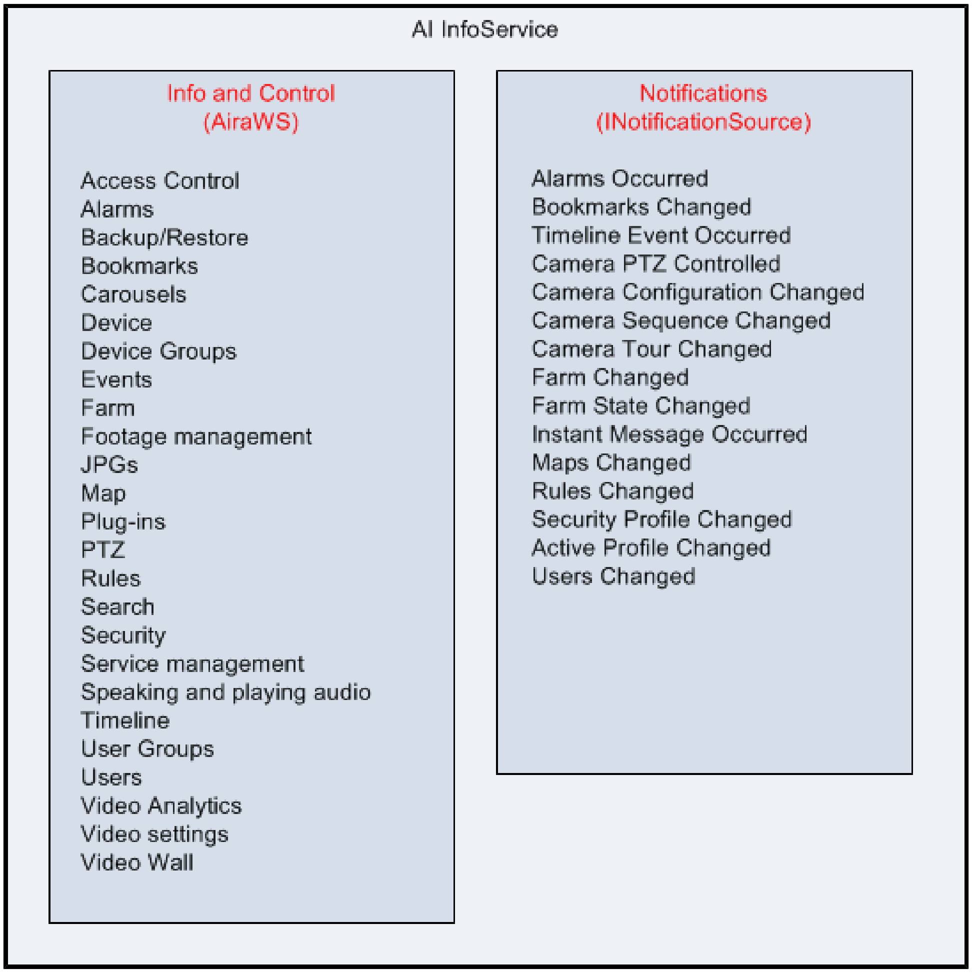
- The managed interfaces must be from executables that target .NET 4.8 or later.
- The recommended IDE is Microsoft Visual Studio 2012 or later.
- The recommended programming language is C#.
- The developer should be familiar with the Senstar Symphony Server.
- It is recommended that you install the Senstar Symphony Server and the Senstar Symphony SDK on different computers.

Portability

It is possible to integrate live video from the Senstar Symphony Server directly to non-Windows platforms using GStreamer and the RESTful Senstar Symphony Mobile Bridge interface.

The portable Senstar Symphony Mobile Bridge interface is not as full featured as the Windows accessible SeerWS interface. If you need functionality in SeerWS from non Windows platforms, then you must develop your own layer that provides this interface. Ported implementations of .NET will work (such as Mono), but the referenced .NET libraries link to Windows style DLLs that require deeper platform equivalence.

Server API overview



AI InfoService

Info and Control
(AiraWS)

Access Control
Alarms
Backup/Restore
Bookmarks
Carousels
Device
Device Groups
Events
Farm
Footage management
JPGs
Map
Plug-ins
PTZ
Rules
Search
Security
Service management
Speaking and playing audio
Timeline
User Groups
Users
Video Analytics
Video settings
Video Wall

Notifications
(INotificationSource)

Alarms Occurred
Bookmarks Changed
Timeline Event Occurred
Camera PTZ Controlled
Camera Configuration Changed
Camera Sequence Changed
Camera Tour Changed
Farm Changed
Farm State Changed
Instant Message Occurred
Maps Changed
Rules Changed
Security Profile Changed
Active Profile Changed
Users Changed

Sample code

Senstar Symphony Server farm connectivity

Use cases	SDK projects
Connect to a Senstar Symphony Server farm	NotificationMonitorExample FarmTest

Video

Use cases	SDK projects
Get a list of cameras	LiveStreamTest FarmTest
Connect and render live video	LiveStreamTest
Connect and render historical video	LiveStreamTest
Get a historical JPG	AlarmHandlerExample
Export video	ExportVideo

PTZ cameras

Use cases	SDK projects
Control PTZ on a video device	LiveStreamTest FarmTest
Load guard tour locations	LiveStreamTest FarmTest
Go to a specific guard tour location	LiveStreamTest FarmTest

Events

Use cases	SDK projects
Trigger an alarm	AddAlarmToCamera FarmTest
Listen for alarms	AlarmHandlerExample MobileBridge

Use cases	SDK projects
Get a list of alarms	LiveStreamTest
Mark an alarm	LiveStreamTest
Listen for farm events	NotificationMonitorExample NotificationMonitors MobileBridge

Farm settings

Use cases	SDK projects
Retrieve farm settings	WebService
Execute a web service method	WebServiceConsole

Recording

Use cases	SDK projects
Start/stop recording	LiveStreamTest FarmTest

Senstar Symphony Client control

Use cases	SDK projects
Switch cameras in the Senstar Symphony Client interface using the Senstar Symphony Server API	VideoWall FarmTest
Switch cameras in Senstar Symphony Client interface using the Senstar Symphony Client API	VideoWallClient

Reports

Use cases	SDK projects
Get a heatmap	GetDensityImageJpg

Security

Use cases	SDK projects
Change the currently active security profile	SetActiveProfile
Add/remove users	UserManagement

Video analytics

Use cases	SDK projects
Get live XML metadata from a running analytics engine	XMLStream
Analyze video in Senstar Symphony from a third-party video system	AlgorithmIntegration
Load a Senstar Symphony UI component in a third-party application	PluginHost
Incorporate your own analytics engine	AlgoSample
Incorporate your own rules engine	AlgoSample TestRuleEngine
Incorporate your own analytics engine configuration	AlgoSampleConfiguration
Incorporate your own rules engine configuration	AlgoSampleConfiguration

Timeline

Use cases	SDK projects
Get timeline information	TimelineGetter

Navigation

Use cases	SDK projects
Use navigation buttons to control video	FarmTest

DIO rules

Use cases	SDK projects
Import/export DIO rules from/to a CSV file	AddDIORulesFromCSV ExportDIORulesToCSV

Hardware devices

Use cases	SDK projects
Get a list of all hardware devices	PSA
Add a new hardware device	PSA

Camera usage

Use cases	SDK projects
View camera-use information	ClientBandwidth

Managed libraries

The Senstar Symphony SDK uses the following managed libraries:

Component	Description
`Seer.Connectivity.dll`	• Library for connecting to the Senstar Symphony Server (usually used indirectly by samples)
`Seer.BaseLibCS.dll`	• `Seer.BaseLibCS.Communication.ServerConnectionManager` for client-server connectivity (usually used indirectly by samples) • `Seer.BaseLibCS.AiraWS.Signals` for Web service interface • `Seer.BaseLibCS.CameraMessageStruct` for structure definitions • `Seer.BaseLibCS.Utils.MulticastMessage` for constants
`Core.dll`	• Source part of the Senstar Symphony SDK • `Seer.SDK.SDKFarm` for farm connectivity and event stream notifications
`Seer.Farm.dll`	• `Seer.Farm.DeviceModel.Client` for client-side device operations and information • `Seer.Farm.DeviceModel.Server` for server-side device operations and information
`Internationalization.dll`	• `Seer.Internationalization.Translator` for language translation into all supported languages
`NotificationMonitors.dll`	• Source part of the Senstar Symphony SDK • Interface to the event stream that comes from the Senstar Symphony Server

The following managed libraries are dependencies of the above libraries and likely do not need to be used by your code directly. However, you might need to add them as references to your project. If they do need to be added, the compiler will fail with a message stating what reference needs to be added.

- `Seer.Common.dll`
- `DeviceModel.Dio.dll`
- `DeviceModel.Security.dll`
- `Farm.Security.dll`
- `SecurityLib.dll`

The following managed libraries enable access to video streaming.

- `AxVideoRecvCtrl.dll` wraps `VideoRecvCtrl.dll`
- `BaseIDL.dll` is a set of COM interfaces that aid in streaming video

Media-streaming libraries

Component	Description
`AxVideoRecvCtrl`	• ActiveX control that supports the Dispatch interface • Can be used by simple applications such as VB or Web-based applications
`IVideoRecvCtrl`	• A COM interface exposed by `AxVideoRecvCtrl` • Supports richer API when combined with `INetworkEndpoint` and `IHistoricalSeek` • This interface is demonstrated by the LiveStreamTest sample • This is the recommended interface
`OCXDll.dll`	• Library that allows users ti dynamically create, use, and destroy `AxVideoRecvCtrl`

Persistence

Files

Files	Description
_FootageArchive (`*.dat`)	• Video • Audio • Overlays (decorations)
_Signals (`*.dat`)	• Timeline data
_Signals2 (`*.dat`)	• Metadata (usually from video analytics)
Exported movie files (`*.aira`)	• Video • Audio • Overlays (decorations)

Database

Data	Description
Settings	• Most settings (except users, groups, and servers)
Servers	• All servers and server states in a farm
Alarms	• All alarms
Senstar Xnet data	• License information
ObjectCounts	• Count of objects per minute (video analytics)
LineCounts	• Count of line crossings per minute (video analytics)

You can use `Dbupdater.exe` from the command line to run SQL against the database.

```
dbupdater "select top 10 * from settings order by sequenceid desc"
```

Shows the last ten settings that have changed.

```
dbupdater "select * from servers"
```

Shows all server addresses and states in the farm.

```
dbupdater "select count(*) from alarms"
```

Shows the current count of all alarms.

All database times are UTC encoded as the number of seconds (or milliseconds) since 1970. Run `dbconfiguration.exe` to view or change the current database connection string.

Add streaming video

You can use the Senstar Symphony SDK to add streaming video to a project.

1. Open the design view of the form to which you want to add the control.
2. From the **Tools** menu, click **Add/Remove Toolbox Items**.
3. Click the **COM Components** tab.
4. Select the **VideoRecvCtrl Control** box.
5. Click **OK**.

   The ActiveX control appears in the toolbox in the **Windows Forms** section.
6. Drag the ActiveX control onto the design view.

Enable panel switching

You can use the Senstar Symphony SDK to enable panel switching in the Senstar Symphony Client interface.

1. Perform one of the following tasks:
   - To switch video panels using the Senstar Symphony Server, use the FarmTest or VideoWall application.
   - To switch video panels using the Senstar Symphony Client, use the VideoWallClient application.
2. Ensure that the user has Video Wall Change Panel permissions.
3. To register the Senstar Symphony Client in which the video panel switch will occur as a video wall client, complete the following steps:
   a) In the Senstar Symphony Client interface, click **Settings** > **Video Wall**.
   b) Click the **Video Wall Client Configuration** tab.
   c) Click **Register current Symphony Client**.
4. To get the panel name, complete the following steps:
   a) In the Senstar Symphony Client interface, right-click the panel and click **Settings**.
   b) Click the **Child** tab to display the panel name.

      If you use the VideoWall application in the command prompt and the panel name includes a space, surround the panel name with double quotes (e.g., "*panel name*").
5. Use one of the following IP addresses:
   - In the FarmTest and VideoWall applications, use the IP address of the Senstar Symphony Server.
   - In the VideoWallClient application, use the IP address of the Senstar Symphony Client.

Get an image

You can use the Senstar Symphony SDK to get a JPEG image from a project.

1. Use the `GetJPEG` command to generate a JPEG image and return its URL.
2. Use the `GetJPEGImage` command to get the byte array of a JPEG image for a specific time and camera.
3. Use the `GetJPEGImage3` command to extend the `GetJPEGImage` command and add parameters for the decorations.
4. Use the `GetJPEGImage4` command to extend the `GetJPEGImage3` command and add parameters for the font used on the image.

Decoration options

The following parameters determine the decorations that JPEG images include.

```
DecorationOptions options = new DecorationOptions(15, 0, true, 0, true);
// Pass decorations. EncodedDecorationOptions to GetJPEG* functions

// You will want to use the following DecorationOptions constructor
public DecorationOptions(int decorations, int dateFormat, bool in24HourNotation, uint
 streamIndex, bool canViewPrivateVideo)
{
    // BITWISE OR together the following to select decoration
    // 1 = rectangles.
    // 2 = messages.
    // 4 = time.
    // 8 = paths.
    _rawDecoration = decorations;

    // SET to one of the following
    // 0 = yyyy/MM/dd
    // 1 = dd/MM/yyyy
    // 2 = MM/dd/yyyy
    _dateFormat = dateFormat;

    // True for 24 hour time format. False for 12 hour time format.
    _in24HourNotation = in24HourNotation;

    // 0 based stream number. The Symphony Client UI is 1 based.
    _streamIndex = streamIndex;
}
```

Activate a relay

To activate or deactivate a relay, call the following Web service method:

```
public void PerformAction(string username, string password, string sActionReq)
```

To activate a relay, `sActionReq` uses the following format:

```
<action><ServerIP>10.234.8.30</ServerIP><camera><On Relay='1' DeviceID='2'/></camera></action>
```

To deactivate a relay, `sActionReq` uses the following format:

```
<action><ServerIP>10.234.8.30</ServerIP><camera><Off Relay='1' DeviceID='2'/></camera></action>
```

Enable a camera tour

To enable or disable a camera tour, use the following SQL command:

```
declare @xml xml, @cameraId int, @tourName varchar(250), @disabled int;
set @cameraId = 3;
set @tourName = 'Camera Tour 2';
set @disabled = 1;
select @xml = CAST(v as xml) from settings where Type = 'Camera' and Section = 'Camera' and K =
 'cameraTour' and ID = @cameraId;
set @xml.modify('replace value of (/TourGroup/cameraTour[@name=sql:variable(''@tourName'')]/
disable/text())[1] with sql:variable(''@disabled'')');
update Settings set V = CAST(@xml as nvarchar(max)) where Type = 'Camera' and Section = 'Camera'
 and K = 'cameraTour' and ID = @cameraId;
```

To run a batch file on the Senstar Symphony Server, pass the following command to the DB Updater utility:

```
dbupdater "declare @xml xml, @cameraId int, @tourName varchar(250), @disabled int;
set @cameraId = 3;
set @tourName = 'Camera Tour 2';
set @disabled = 1;
select @xml = CAST(v as xml) from settings where Type = 'Camera' and Section = 'Camera' and K =
 'cameraTour' and ID = @cameraId;
set @xml.modify('replace value of (/TourGroup/cameraTour[@name=sql:variable(''@tourName'')]/
disable/text())[1] with sql:variable(''@disabled'')');
update Settings set V = CAST(@xml as nvarchar(max)) where Type = 'Camera' and Section = 'Camera'
 and K = 'cameraTour' and ID = @cameraId;"
```

Integrate a third-party video analytic

You can use sample projects in the Senstar Symphony SDK to integrate a third-party video analytic with the Senstar Symphony Server.

Every camera connected to the Senstar Symphony Server has its own AI Tracker service (`Trackerapp.exe`). The video from the camera is decompressed and passed into AI Tracker filter (`Tracker.ax`), which contains the analytic and rule engines. The analytic engine processes the video and produces XML metadata that is passed to the rules engine. The rules engine processes the metadata and creates alarms in the Alarm database table.

When a user configures an analytic or rule in the Senstar Symphony Server configuration interface, the Senstar Symphony Server saves the configurations in the database. The XML strings are used to initialize the analytic and rule engines when the AI Tracker service starts. The following is an example of the XML:

```
<State><DisplayText>% has been changed</DisplayText><DisplayPercentage>1</DisplayPercentage></
State>
```

In addition to the XML strings, there are some properties used to describe the analytic to the Senstar Symphony Server. The following fields are of special note:

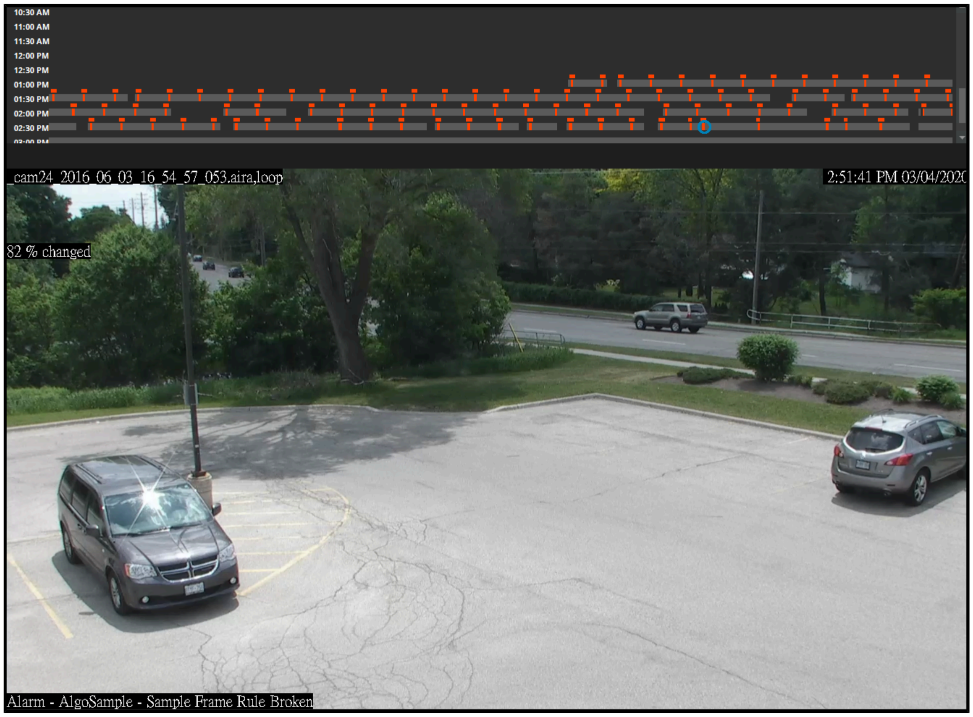
- Name: the name of the analytic displayed in the Senstar Symphony Server configuration interface
- Description: the description of the analytic displayed in Senstar Symphony Server configuration interface
- Configurable: whether the analytic has Web interface for configuration
- ConfigurableForRules: whether the analytic has a Web interface for rule configuration
- SearchSupported: flag to enable/disable searching by the metadata created by the analytic

There are two projects in the Senstar Symphony SDK that provide a sample video analytic that you can integrate with the Senstar Symphony Server:

- `AlgoSample`: the video analytic and rule engine project
- `ConfigureWeb_AlgoSample`: the video analytic and rule configuration project

If you change the names of the sample projects, the name of the configuration project must start with `ConfigureWeb_` for the Senstar Symphony Server to recognize the project.

The `AlgoSample` video analytic counts the number of frames that change and displays either the count or the percentage as a decoration.



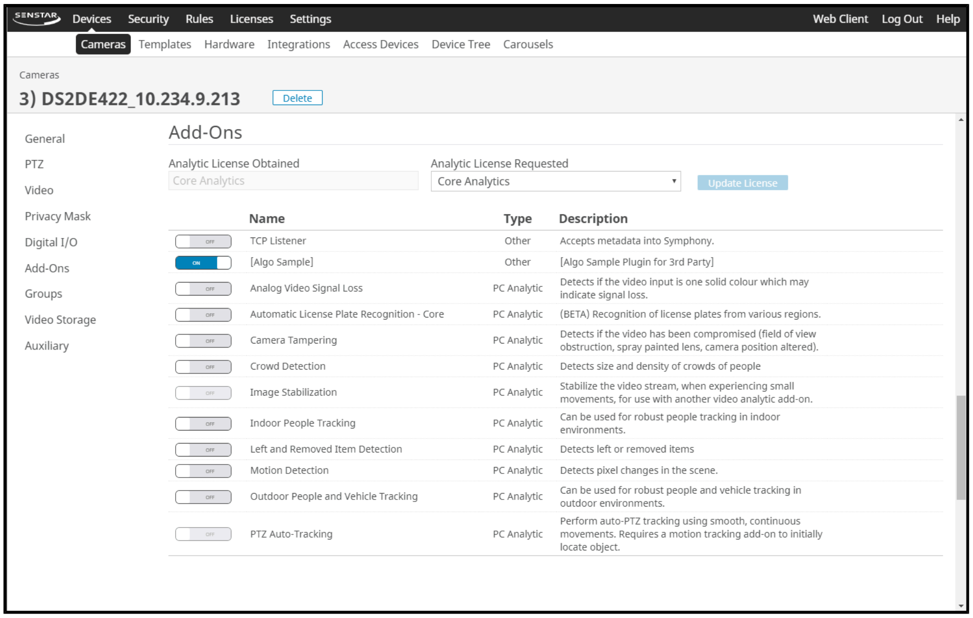
_cam24_2016_06_03_16_54_57_053.aira.loop

2:51:41 PM 03/04/2020

82 % changed

Alarm - AlgoSample - Sample Frame Rule Broken

1. Download the newest Senstar Symphony SDK.
2. On the computer that hosts the Senstar Symphony Server, stop all of the Senstar Symphony services (`killall 9`).
3. Copy the `AlgoSample.*` and `ConfigureWeb_AlgoSample.*` files from the `SDK\bin\algos` folder to the `...\Senstar\Symphony SDK\bin\algos\` folder on the computer that hosts the Senstar Symphony Server.
4. On the computer that hosts the Senstar Symphony Server, start all of the Senstar Symphony services (`killall 5`).
5. In the Senstar Symphony Server configuration interface, configure the a camera to use the `AlgoSample` project as a video analytic.



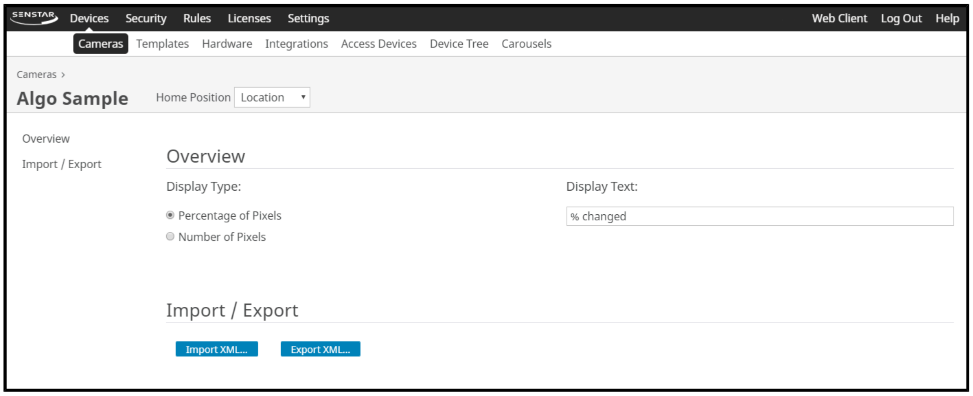
3) DS2DE422_10.234.9.213

Add-Ons

Analytic License Obtained: Core Analytics

Analytic License Requested: Core Analytics

	Name	Type	Description
OFF	TCP Listener	Other	Accepts metadata into Symphony.
ON	[Algo Sample]	Other	[Algo Sample Plugin for 3rd Party]
OFF	Analog Video Signal Loss	PC Analytic	Detects if the video input is one solid colour which may indicate signal loss.
OFF	Automatic License Plate Recognition - Core	PC Analytic	(BETA) Recognition of license plates from various regions.
OFF	Camera Tampering	PC Analytic	Detects if the video has been compromised (field of view obstruction, spray painted lens, camera position altered).
OFF	Crowd Detection	PC Analytic	Detects size and density of crowds of people
OFF	Image Stabilization	PC Analytic	Stabilize the video stream, when experiencing small movements, for use with another video analytic add-on.
OFF	Indoor People Tracking	PC Analytic	Can be used for robust people tracking in indoor environments.
OFF	Left and Removed Item Detection	PC Analytic	Detects left or removed items
OFF	Motion Detection	PC Analytic	Detects pixel changes in the scene.
OFF	Outdoor People and Vehicle Tracking	PC Analytic	Can be used for robust people and vehicle tracking in outdoor environments.
OFF	PTZ Auto-Tracking	PC Analytic	Perform auto-PTZ tracking using smooth, continuous movements. Requires a motion tracking add-on to initially locate object.



Algo Sample

Home Position: Location

Overview

Display Type:

● Percentage of Pixels

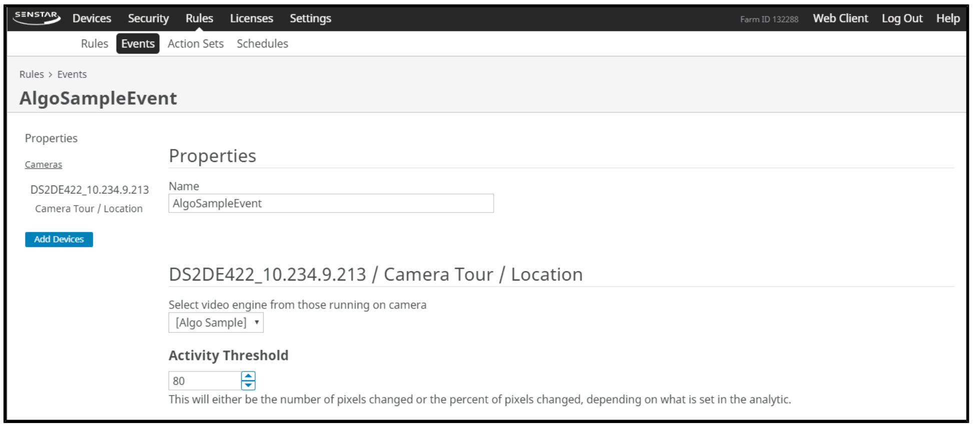
○ Number of Pixels

Display Text: % changed

Import / Export

6. Test the `AlgoSample` video analytic.
7. In the Senstar Symphony Server configuration interface, create an event for the `AlgoSample` video analytic that triggers a rule.

   The `AlgoSample` video analytic should trigger an alarm when the number or percentage goes above a threshold that you set in the sample code.



AlgoSampleEvent

Properties

Name: AlgoSampleEvent

DS2DE422_10.234.9.213 / Camera Tour / Location

Select video engine from those running on camera: [Algo Sample]

Activity Threshold: 80

This will either be the number of pixels changed or the percent of pixels changed, depending on what is set in the analytic.

8. Modify and compile the `AlgoSample` and `ConfigureWeb_AlgoSample` projects to integrate the third-party video analytic.

   You must compile the `AlgoSample` and `ConfigureWeb_AlgoSample` projects in Release mode for the `AlgoSample` video analytic to work with the Senstar Symphony Server.
9. Repeat steps 2 to 5 for the new `AlgoSample` and `ConfigureWeb_AlgoSample` projects.

SenstarIntelligenceStream-2.0.xml

```
<?xml version="1.0"?>

<stream
    xmlns="http://www.senstar.com"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="SenstarIntelligenceStream-2.0.xsd">

 <frame
  utc='11051078632640000'
  brightness='10'
  contrast='10'
  cordcut='1'
  cameraobstructed='1'>

  <!-- xml for creating a new object or tracking one already created -->
  <object id='124' confidence='95'>
   <class>Car</class>
   <rect x='0' y='0' width='192' height='475'/>
   <speed>12.6</speed>
  </object>
  <!-- an object which came to rest in the video -->
  <!-- if the object was actively tracked before stopping, id will correspond to the tracked
 object -->
  <object id='122'>
   <class>Car</class>
   <rect x='45' y='102' width='150' height='50'/>
   <speed>0</speed>
   <state>Stationary</state>
  </object>

  <!-- xml for deleting an object which was tracked up until now -->
  <!-- "id" must have appeared at least once in an object tag -->
  <delete id='123'/>

  <!-- xml for flagging a policy break -->
  <!-- if the alarm was caused by an object moving or a stationary object -->
  <!-- the id will correspond with the offending object -->
  <!-- if the alarm was a cord-cut or camera-obstructed alarm, the id will be 0 -->
  <alarm id='123'>
   <policyid>1</policyid>
   <alarmonid>2</alarmonid>
   <occur>0</occur>
  </alarm>
 </frame>
 <frame
  utc='11051078632641000'
  brightness='10'
  contrast='10'
  >
  <!-- xml for deleting an object which was tracked up until now -->
  <delete id='124'/>

  <!-- xml for deleting an object which was stationary up until now -->
  <delete id='122'/>
 </frame>
</stream>
```

SenstarIntelligenceStream-2.0.xsd

```
<?xml version="1.0"?>
<xs:schema    xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.senstar.com"
  xmlns="http://www.senstar.com">


 <xs:simpleType name="classtype">
  <xs:restriction base="xs:string">
   <xs:enumeration value="Car"/>
   <xs:enumeration value="Person"/>
   <xs:enumeration value="Unknown"/>
  </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="percentage">
  <xs:restriction base="xs:decimal">
   <xs:minInclusive value="0"/>
   <xs:maxInclusive value="100"/>
  </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="statetype">
  <xs:restriction base="xs:string">
   <xs:enumeration value="Stationary"/>
   <xs:enumeration value="Leaving"/>
  </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="colourvaluetype">
  <xs:restriction base="xs:string">
   <xs:pattern value="#[0-9a-fA-F]{6}"/>
  </xs:restriction>
 </xs:simpleType>

 <xs:element name="colour">
  <xs:complexType>
   <xs:simpleContent>
    <xs:extension base="xs:string">
     <xs:attribute type="colourvaluetype" name="value" use="required"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
 </xs:element>

 <xs:element name="colourlist">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="colour" minOccurs="1" maxOccurs="16"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>

 <xs:element name="class">
  <xs:complexType>
   <xs:simpleContent>
    <xs:extension base="classtype">
     <xs:attribute name="confidence" type="percentage"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
 </xs:element>

 <xs:element name="rect">
  <xs:complexType>
   <xs:attribute name="x"  type="xs:integer"/>
   <xs:attribute name="y"  type="xs:integer"/>
   <xs:attribute name="width" type="xs:integer"/>
   <xs:attribute name="height" type="xs:integer"/>
  </xs:complexType>
 </xs:element>

 <xs:element name="object">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="class"/>
    <xs:element ref="rect"/>
    <xs:element name="speed"  type="xs:decimal"/>
    <xs:element name="size"   type="decimal"/>
    <xs:element name="state"  type="statetype"/>
    <xs:element ref="colourlist"/>
   </xs:sequence>
   <xs:attribute name="id"      type="xs:integer" use="required"/>
   <xs:attribute name="confidence"  type="percentage" default="0"/>   <!-- Our confidence that
 this object should be tracked as a foreground object 0 == definitely distracting motion, 100 ==
 definitely foreground -->
   <xs:anyAttribute/>
  </xs:complexType>
 </xs:element>

 <xs:element name="alarm">
  <xs:complexType>
   <xs:attribute name="id"      type="xs:integer" use="required"/>
   <xs:attribute name="policyid" type="xs:integer" use="required"/>
   <xs:attribute name="alarmonid"  type="xs:integer" use="required"/>
   <xs:attribute name="occur"  type="xs:unsignedLong"/>
   <xs:anyAttribute/>
  </xs:complexType>
 </xs:element>

 <xs:element name="delete">
  <xs:complexType>
   <xs:attribute name="id"   type="xs:integer" use="required"/>
  </xs:complexType>
 </xs:element>


 <xs:element name="stream">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="frame">
     <xs:complexType>
      <xs:sequence>
       <xs:element ref="object"    minOccurs="0" maxOccurs="65535"/>
       <xs:element ref="delete"    minOccurs="0" maxOccurs="65535"/>
       <xs:element ref="alarm"    minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
```

```
      <xs:attribute name="utc" type="xs:unsignedLong"/>
      <xs:attribute name="brightness" type="xs:integer"/>
      <xs:attribute name="contrast" type="xs:integer"/>
      <xs:attribute name="cordcut" type="xs:boolean" default="0"/>
      <xs:attribute name="cameraobstructed" type="xs:boolean" default="0"/>
      <xs:anyAttribute/>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Legal information

This material is for informational purposes only. Senstar makes no warranties, express, implied or statutory, as to the information in this document.

Senstar may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Senstar, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Senstar and the Senstar logo are registered trademarks of Senstar Corporation.

All other trademarks are the property of their respective owners.