



AnalyticVideo Server
Video Analytics
User Guide
version 1.2

Copyright © 2013 Institute of Digital and Optical Technologies B. V.

All rights reserved

Contents

CONTENTS	2
1 INTRODUCTION	3
2 RULE DESCRIPTION	4
3 SUPPORTED RULES	6
3.1 CORRESPONDENCE BETWEEN AVS RULES AND ONVIF RULE TYPES.....	6
3.2 LINEDETECTOR ONVIF RULE	8
3.3 COUNTAGGREGATION ONVIF RULE.....	9
3.4 DECLARATIVEMOTIONDETECTOR ONVIF RULE	10
4 CONTACT US	12

1 INTRODUCTION

AnalyticVideo Server is divided into image analysis and application-specific parts. The interface between these two parts produces an abstraction that describes the scene based on the objects present. The application specific part performs a comparison of the scene descriptions and of the scene rules (such as virtual lines that are prohibited to cross, or polygons that define regions of interest). Other rules may represent intra-object behavior such as objects following other objects (to form a tailgating detection). Such rules can also be used to describe prohibited object motion, which may be used to establish a speed limit.

These two separate parts, referred to as the [Video Analytics Engine](#) and as the [Rule Engine](#), together with the events, form the video analytics architecture.

The output from the [Video Analytics Engine](#) is called a [Scene Description](#). The [Scene Description](#) represents the abstraction of the scene in terms of the objects, either static or dynamic, that are part of the scene. This specification defines an XML-based [Scene Description Interface](#) including data types and data transport mechanisms.

Rules describe how the [Scene Description](#) is interpreted and how to react on that information. The specification defines standard rule syntax and methods to communicate these rules from the application to the device.

The configuration of a rule has two required attributes: one specifies the name and the other specifies the type of the rule. The different configuration parameters are listed below the parameters element of the rule element. Each parameter is either a [SimpleItem](#) or an [ElementItem](#). Both [ItemDescriptions](#) contain a name attribute to identify the parameter and a [Type](#) attribute to reference a specific XML schema type. In case of the [SimpleItemDescription](#), the type attribute shall reference a [SimpleType](#) schema definition. In case of the [ElementItemDescription](#), the [Type](#) attribute shall reference a global element declaration of an XML schema. The name attribute of each item shall be unique within the parameter list.

2 RULE DESCRIPTION

The description of a rule contains the type information of all parameters belonging to a certain rule type and the description of the output produced by such a rule. The output of the [Rule Engine](#) is Events which can either be used in an Event Engine or be subscribed to by a client.

The definitions are included for convenience:

```
<xs:element name="RuleDescription" type="tt:ConfigDescription"/>
<xs:complexType name="ConfigDescription">
  <xs:sequence>
    <xs:element name="ParameterDescription"
      type="tt:ItemListDescription"/>
    <xs:element name="MessageDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tt:MessageDescription">
            <xs:sequence>
              <xs:element name="ParentTopic" type="xs:string"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    ...
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="ItemListDescription">
  <xs:sequence>
    <xs:element name="SimpleItemDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
        <xs:attribute name="Type" type="xs:QName" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ElementItemDescription" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
        <xs:attribute name="Type" type="xs:QName" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The available rules can be retrieved via [GetRules](#). Pass unique module names that can be later used as a reference. The parameters of the rules must match those of the corresponding description.

GetRules

Description: List the currently assigned set of rules of a VideoAnalyticsConfiguration.

SOAP action: <http://www.onvif.org/ver20/analytics/wsd/GetRules>

Input: [GetRules]

- **ConfigurationToken** [ReferenceToken]
Reference to an existing VideoAnalyticsConfiguration.

Output: [GetRulesResponse]

- **Rule** - optional, unbounded; [Config]
 - **Parameters** [ItemList]
List of configuration parameters as defined in the corresponding description.
 - **SimpleItem** - optional, unbounded;
Value name pair as defined by the corresponding description.
 - **Name** - required; [string]
Item name.
 - **Value** - required; [anySimpleType]
Item value. The type is defined in the corresponding description.
 - **ElementItem** - optional, unbounded;
Complex value structure.
 - **xsd:any**
XML tree containing the element value as defined in the corresponding description.
 - **Name** - required; [string]
Item name.
 - **Extension** - optional; [ItemListExtension]
 - **Name** - required; [string]
Name of the configuration.
 - **Type** - required; [QName]
Type of the configuration represented by a unique QName. The Type characterizes a ConfigDescription defining the Parameters.

3 SUPPORTED RULES

3.1 Correspondence between AVS rules and ONVIF rule types

For now, *AnalyticVideo Server* includes eight video analytic rules:

1. [Lane Object Counter](#) — counts moving objects in the specified zones. Suits perfectly for counting cars on the lanes. For better use, select each lane as a zone of interest. Sends event on object counter change.
2. [Direction Detector](#) — sends event when an object moves in illegal direction. Double-click to open the angle settings to select permitted direction of movement.
3. [In Zone Counter](#) — counts objects located in the specified zones of interest at the moment. Sends event on object counter change.
4. [Line Crossing](#) — sends event when an object crosses the specified line in the specified direction. Pressing [Line tools](#) allows drawing lines for [Line Crossing](#) module:
 - a. [One Direction Tool](#) - counts objects that intersect the line in one direction;
 - b. [Both Direction Tool](#) - counts objects that intersect the line in both directions.
5. [Object Counter](#) — counts objects that passed through the specified zones of interest. It sends event on counter change. Nature of work is similar to [Lane Object Counter](#) plugin. The difference is that [Lane Object Counter](#) plugin runs one detector in one zone of interest. [Object counter](#) plugin runs one detector in all zones of interest. Therefore, when zones of interest intersect, use of [Lane Object Counter](#) is preferred.
6. [Simple Tracking](#) — turns on object detection and tracking algorithms. Press [Set event interval](#) to setup heartbeat interval of events. Events consist of data about all objects position, size and type ([Determine Human/Car Objects](#) option has to be ON).
7. [Thread Tracking](#) — draws the moving objects trajectories. This plugin does not send ONVIF events. Press [Settings](#) to show the dialog for [Thread Tracking](#) plugin with additional configuring parameters:
 - a. [Object Radius](#) — object track is shown as a fixed length line;
 - b. [Frame Number](#) — object track is shown by curve which length depends on the frame number.

8. **Unattended Objects Detector** — enables automatic detection of appeared or disappeared objects on a scene. Sends events consist of data about all unattended objects position, size and type (**Determine Human/Car Objects** option has to be **ON**).

Table shows correspondence between *AnalyticVideo Server* rules and ONVIF rule types. For example, **Direction Detector** plugin, that has two zones of interest, sends ONVIF events that have **CountAggregation** type and **DD_1** or **DD_2** names.

Correspondence between <i>AnalyticVideo Server</i> rules and ONVIF rule types	
Rule name	Rule type
LC_xxx (Line Crossing)	LineDetector
LOC_xxx (Lane Object Counter)	CountAggregation
DD_xxx (Direction Detector)	CountAggregation
IZC_xxx (In Zone Counter)	CountAggregation
OC_xxx (Object Counter)	CountAggregation
MD (Simple Tracking)	DeclarativeMotionDetector
NOMOD (Unattended Object Detector)	DeclarativeMotionDetector

In the rule name **xxx** is a region (zone) of interest number.

The **input parameters** of a certain rule types are listed below the **ParameterDescription** element. The **output** produced by this rule type is described in multiple **MessageDescription** elements.

A device supporting **ONVIF Video Analytics Service** shall implement the **Scene Description Interface** and allow events to be dispatched using the **Event Service**. If the device additionally supports a **Rule Engine** then it shall implement the **Rules Analytics Modules Interface**.

The structure of the **Message** payload contains three groups: **Source**, **Key**, and **Data**. Each group contains a set of **Simple** and **ElementItems**. For each topic, a device can describe which item will be part of a notification produced by this topic using a message description language.

3.2 LineDetector ONVIF Rule

The [LineDetector](#) is defined by a non-intersecting simple polyline. If an [Object](#) crosses the polyline in the specified direction, the [Rule Engine](#) sends a [Crossed](#) event containing the name of the [LineDetector](#) and a reference to the object which has crossed the line. As directions, one can select between [Left](#), [Right](#), and [Any](#), where directions [Left](#) and [Right](#) refer to the direction walking along the line from the first point to the second point and are the prohibited directions.

The [LineDetector](#) resembles the following code using the [Rule Description Language](#):

```
<tt:RuleDescription Name="tt:LineDetector">
  <tt:Parameters>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction" />
    <tt:ElementItemDescription Name="Segments" Type="tt:Polyline" />
  </tt:Parameters>
  <tt:MessageDescription>
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
        Type="tt:ReferenceToken" />
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
        Type="tt:ReferenceToken" />
      <tt:SimpleItemDescription Name="Rule" Type="xs:string" />
    </tt:Source>
    <tt>Data>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer" />
    </tt>Data>
    <tt:ParentTopic>tns1:RuleEngine/LineDetector/Crossed</tt:ParentTopic>
  </tt:MessageDescription>
</tt:RuleDescription>
```

The above rule description defines that a rule instance produces event attached to the topic [tns1:RuleEngine/ LineDetector/Crossed](#), and the configuration parameters:

Input parameters:

Segments	Simple polyline used for detecting that object crosses
Direction	Direction crosses

Output parameters:

ObjectID	Object Identifier
----------	-------------------

3.3 CountAggregation ONVIF Rule

[CountAggregation](#) counts the number of motion object passing through the set of line segments (barrier) and optional direction attribute. The configuration parameters also include the time interval to report the events and time interval to reset its counter.

```
<tt:RuleDescription Name="tt:CountAggregation">
  <tt:Parameters>
  <tt:ElementItemDescription Name="LineSegments"
                            Type="tt:PolylineArrayConfiguration"/>
    <tt:SimpleItemDescription Name="ReportTimeInterval"
Type="xs:duration"/>
    <tt:SimpleItemDescription Name="ResetTimeInterval"
Type="xs:duration"/>
    <tt:SimpleItemDescription Name="Direction" Type="tt:Direction"/>
  </tt:Parameters>

  <tt:MessageDescription IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                              Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                              Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="Count"
Type="xs:nonNegativeInteger"/>
    </tt>Data >
    <tt:ParentTopic>
      tns1:RuleEngine/CountAggregation/Counter
    </tt:ParentTopic>
  </tt:MessageDescription>
</tt:RuleDescription>
```

The above rule description defines that a rule instance produces count event to the topic [tns1:RuleEngine/CountAggregation/Counter](#) and configuration parameters:

Input parameters:

LineSegments	Array of lines used for detecting that object passed all of them for counting
ReportTimeInterval	Time interval to report count information
ResetTimeInterval	Periodic count reset time
Direction	Count direction

Output parameters:

ObjectId	Object Identifier of last counted object
Count	Value of counter

3.4 DeclarativeMotionDetector ONVIF Rule

[DeclarativeMotionDetector](#) detects the object motion against the specified motion attributes. The rule is active within an area (region of interest). The condition is defined against the frame/object element in scene descriptor.

```
<tt:RuleDescription Name="tt:DeclarativeMotionDetector">
  <tt:Parameters>
    <tt:ElementItemDescription Name="Field"
Type="tt:PolygonConfiguration"/>
  </tt:Parameters>
  <tt:MessageDescription IsProperty="true">
    <tt:Source>
      <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
Type="tt:ReferenceToken"/>
      <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
    </tt:Source>
    <tt:Key>
      <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
    </tt:Key>
    <tt>Data>
      <tt:SimpleItemDescription Name="X" Type=" xs:integer "/>
      <tt:SimpleItemDescription Name="Y" Type=" xs:integer "/>
      <tt:SimpleItemDescription Name="W" Type=" xs:integer "/>
      <tt:SimpleItemDescription Name="H" Type=" xs:integer "/>
      <tt:SimpleItemDescription Name="Type" Type=" xs:string "/>
      <tt:SimpleItemDescription Name="Time" Type=" xs:string "/>
    </tt>Data >
    <tt:ParentTopic>
      tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched
    </tt:ParentTopic>
  </tt:MessageDescription>
</tt:RuleDescription>
```

The above rule description defines that a rule instance produces event attached to the topic [tns1:RuleEngine/DeclarativeMotionDetector/MotionMatched](#), and the configuration parameters:

Input parameters:

Field	Defines an area in camera field of view, the motion matching query is applied in this area for detected object
-------	--

Output parameters:

ObjectId	Object Identifier
X & Y	Center of object
W	Object width
H	Object height
Type	Object type
Time	Event time (hh:mm:ss:ms)

4 CONTACT US



Support: support@analyticvideo.com

Licensing: bizdev@analyticvideo.com

Web: www.AnalyticVideo.com

i.Dot

Headquarters

Institute of Digital and Optical Technologies B. V.

Postal address: De Lairesestraat 160-C, 1075 HM
Amsterdam, the Netherlands

Fax: +31.20-8908634

E-mail: bizdev@idotbv.nl

Web: www.idotbv.nl

Copyright © 2013 Institute of Digital and Optical Technologies B. V. All rights reserved.

This guide is for informational purposes only. No warranties, express, implied or statutory, as to the information in this document. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Institute of Digital and Optical Technologies B. V.