milestone

Application note

Metadata Search integration

February 10, 2025

Metadata Search integration

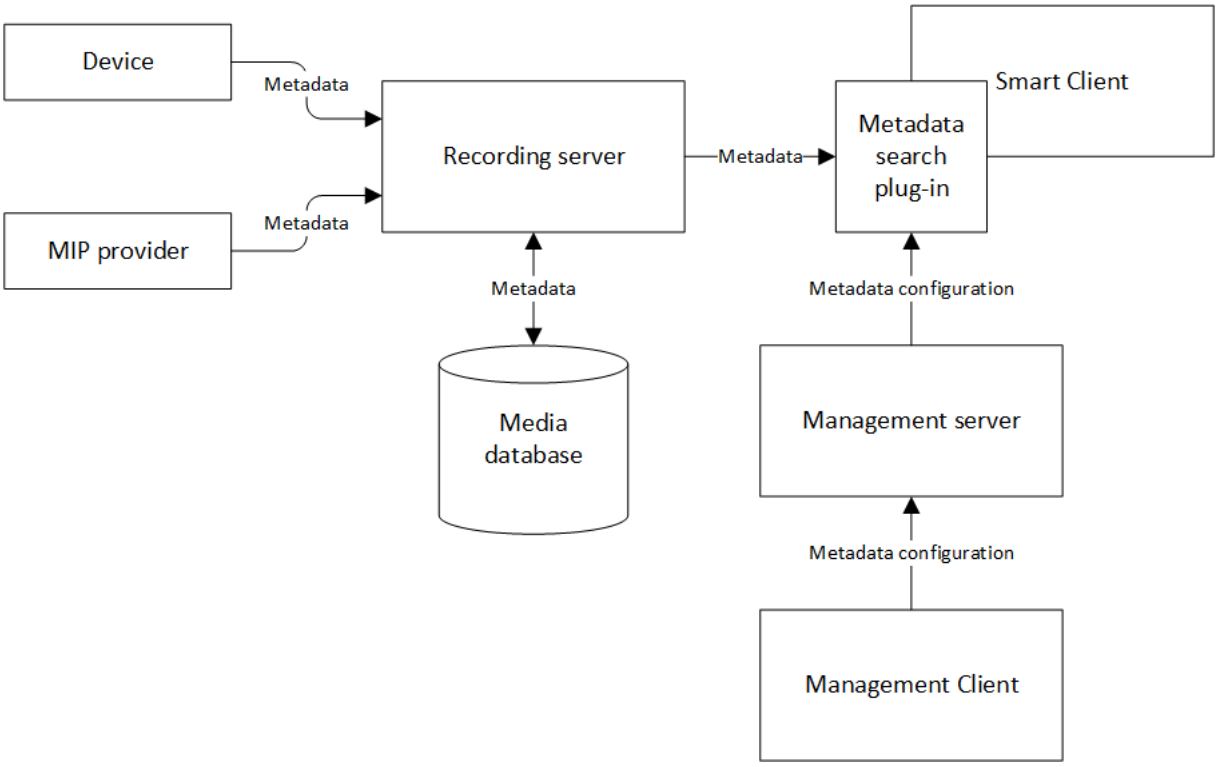
This document is intended to enable Milestone technology partners and Milestone developers to create metadata that will be recognized and available for the Metadata search feature.

Some familiarity with the ONVIF Analytics Service Specification and with XProtect is assumed.

How XProtect VMS works with metadata

Metadata search architecture

This diagram describes the high-level overview of the involved components as well as the flow of metadata.



Metadata can originate from different sources, for example a camera or a MIP-based provider. The recording servers receives the metadata and stores it in media databases. A benefit of storing the metadata alongside the video is that the metadata is exported and deleted along with the video. This simplifies management of the metadata.

In the XProtect Smart Client, a Metadata search plug-in allows the user to define search criteria. The plug-in pulls and filters metadata from the recording servers, and presents the intervals that match the search criteria.

The system needs an understanding of which categories of objects (e.g. people and vehicles) are to available for search. This is configured using the XProtect Management Client and stored on the management server.

Metadata search configuration in XProtect Management Client

> By default, the Metadata search categories and search filters aren't visible in the **Search** tab in XProtect Smart Client.

An XProtect Management Client user with administrator permissions can configure which Metadata search categories and search filters are visible in the XProtect Smart Client.

1. In XProtect Management Client, in the **Site Navigation** pane, select **Metadata Use** > **Metadata Search**.
2. In the **Metadata Search** pane, select the search category that you want to change visibility settings for.
3. To enable the visibility of a search category or search filter, select the corresponding check box. To disable the visibility of a search category or search filter, clear the check box.

Metadata search in XProtect Smart Client

Metadata-based search is integrated in the Smart Client along other categories of search filters, such as alarms, events, motion detection, and bookmarks.

You can think of search categories as a grouping of filters for properties related to video recordings, the frames in the recordings, and objects detected in the frames.

A few examples of how metadata is presented in the Smart Client search feature:

- Objects that are recognized as the class `People` can be categorized by the properties such as `Age`, `Gender`, `Height`, and whether a `Face` was detected. These properties can be used in search filters for the search category People.
- An object can have a `Location`. In addition, the recording also can have a `Location` (the position of the camera itself). Thus, `Location` is a search category by itself, with a filter for geo coordinates (latitude, longitude) and search radius.

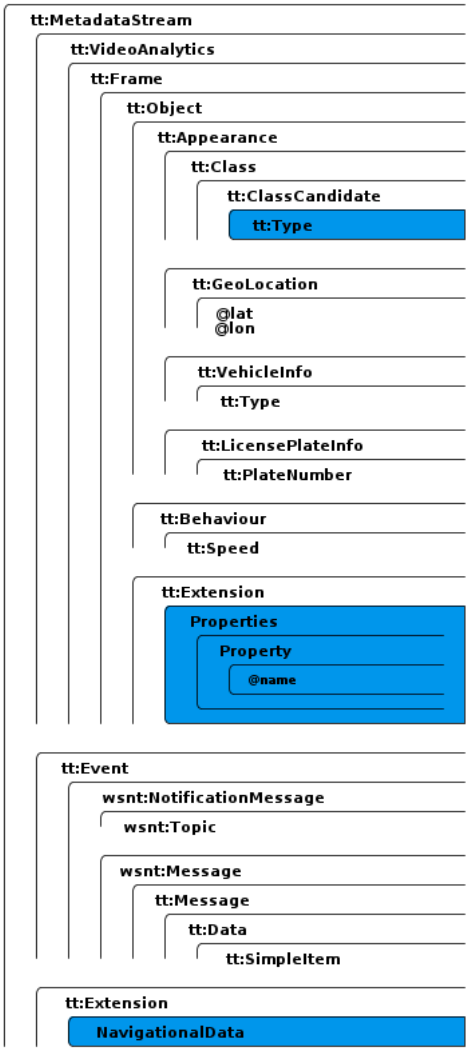
Format

The ONVIF MetadataStream schema

Metadata support in XProtect is based on the ONVIF Analytics Service Specification Version 23.12.

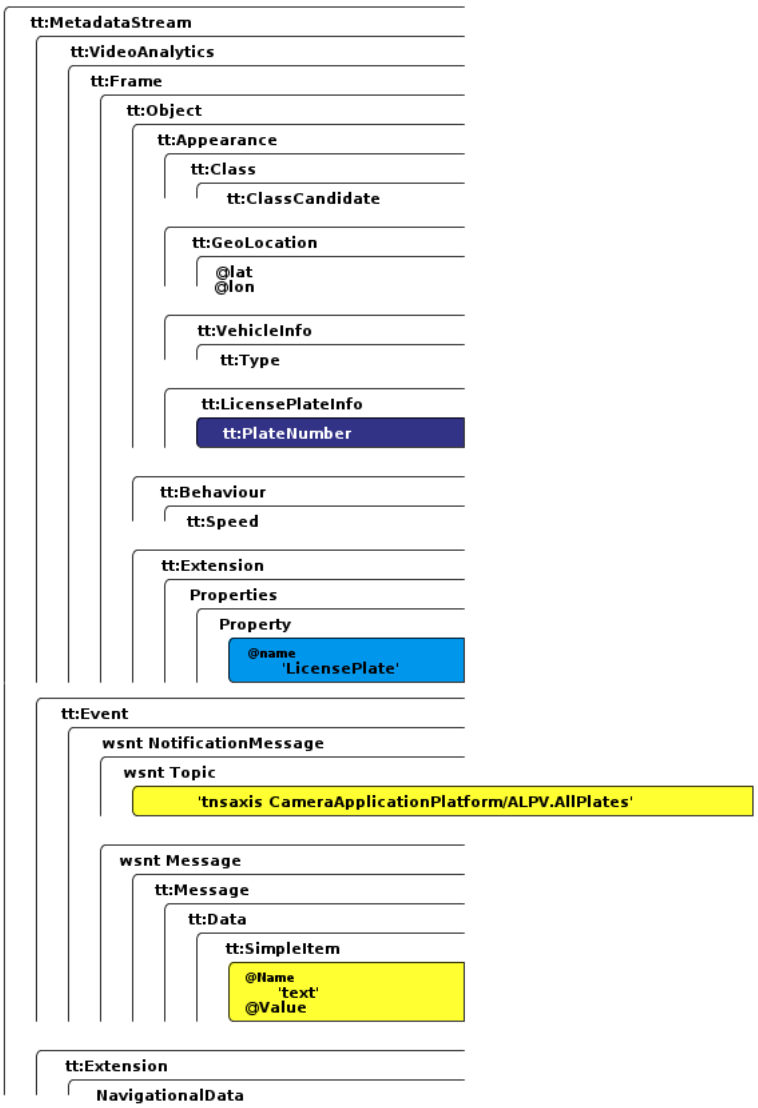
ONVIF schemas are designed to be backwards compatible, forwards compatible, and extensible.

For Metadata search, XProtect extends the ONVIF MetadataStream schema in several places, highlighted in light blue:



At any point in time, there might be several ways to represent the same information, e.g. legacy ONVIF schemas, the current ONVIF schema, and several vendor specific extensions.

For example, for Metadata search, license plate information is found in three places: the default ONVIF representation (dark blue), an XProtect extension (light blue), and another vendor (Axis) extension (yellow).



To accommodate for this variability, XProtect maps ONVIF object classes and object descriptors to XProtect search categories and properties.

The mapping of object classes to search categories is described below in the section Search properties and categories.

ONVIF MetadataStream elements and properties supported by XProtect

Namespace binding

The following prefixes are used:

Prefix	Namespace
bd	http://www.onvif.org/ver20/analytics/humanbody
fc	http://www.onvif.org/ver20/analytics/humanface
tt	http://www.onvif.org/ver10/schema
wsnt	http://docs.oasis-open.org/wsn/b-2

MetadataStream elements not used for Metadata search

XProtect supports ONVIF metadata for several purposes, not just for Metadata search.

Some MetadataStream elements are not available for Metadata search, but can be used e.g. for overlay display in the Smart Client:

- `/tt:MetadataStream/tt:VideoAnalytics/tt:Frame/tt:Object/tt:Appearance/`
  - `tt:Shape/tt:BoundingBox`
  - `tt:Shape/tt:Extension/BoundingBoxAppearance`
  - `tt:Extension/Description`

For more information about these, see Introduction to Metadata.

MetadataStream elements used by Metadata search

Object metadata are extracted from several root XPaths in each `tt:MetadataStream` element. Each of these roots contribute independently to Metadata search.

This table describes the element and attribute values extracted from each root XPath.

The value types refer to the types declared in the ONVIF schema (`metadatastream.xsd` , `humanbody.xsd` , `humanface.xsd`).

> In this table, the namespaces from the ONVIF specification and schemas are used for brevity and clarity. The XPath expressions used by XProtect VMS matches any or no namespace.
>
> For example, `/tt:MetadataStream/tt:VideoAnalytics/tt:Frame/tt:Object/` is matched by the expression `/*[local-name()='MetadataStream']/*[local-name()='VideoAnalytics']/*[local-name()='Frame']/*[local-name()='Object']` .

Root	Element or attribute XPath	Value type	Search property	Search category
`/tt:MetadataStream/tt:VideoAnalytics/tt:Frame/tt:Object/`				
	`../@Colorspace`	`" YCbCr "," RGB "`	Colorspace	
	`tt:Appearance/bd:HumanBody/`			
	`bd:Belonging/bd:Bag`	`node()`	Bag	People
	`bd:Belonging/bd:Bag/bd:Category[text()]`	`bd:KnapsackCategory`	BagCategory	People
	`bd:Belonging/bd:Bag/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	BagColor	People
	`bd:BodyMetric/bd:Height[text()]`	`xs:int` [cm]	HeightOnvif	People
	`bd:Clothing/bd:Bottoms/bd:Category[text()]`	`bd:BottomsCategory`	ClothingBottomCategory	People
	`bd:Clothing/bd:Bottoms/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	ClothingBottomColor	People
	`bd:Clothing/bd:Bottoms/bd:Grain[text()]`	`bd:Grain`	ClothingBottomGrain	People
	`bd:Clothing/bd:Bottoms/bd:Style[text()]`	`bd:BottomsStyle`	ClothingBottomStyle	People
	`bd:Clothing/bd:Gloves/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	GlovesColor	People
	`bd:Clothing/bd:Gloves/bd:Wear[text()]`	`xs:boolean`	Gloves	People
	`bd:Clothing/bd:Scarf/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	ScarfColor	People
	`bd:Clothing/bd:Scarf/bd:Wear[text()]`	`xs:boolean`	Scarf	People
	`bd:Clothing/bd:Shoes/bd:Category[text()]`	`bd:ShoesCategory`	ShoesCategory	People
	`bd:Clothing/bd:Shoes/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	ShoesColor	People
	`bd:Clothing/bd:Tops/bd:Category[text()]`	`bd:TopsCategory`	ClothingTopCategory	People
	`bd:Clothing/bd:Tops/bd:Color/tt:ColorCluster`	`tt:ColorDescriptor`	ClothingTopColor	People
	`bd:Clothing/bd:Tops/bd:Grain[text()]`	`bd:Grain`	ClothingTopGrain	People
	`bd:Clothing/bd:Tops/bd:Style[text()]`	`bd:TopsStyle`	ClothingTopStyle	People

Root	Element or attribute XPath	Value type	Search property	Search category
	tt:Appearance/fc:HumanFace/			
	fc:Accessory/fc:Hat/fc:Color/tt:ColorCluster	tt:ColorDescriptor	HatColor	People
	fc:Accessory/fc:Hat/fc:Wear[text()]	xs:boolean	Hat	People
	fc:Accessory/fc:Helmet/fc:Color/tt:ColorCluster	tt:ColorDescriptor	HelmetColor	People
	fc:Accessory/fc:Helmet/fc:Wear[text()]	xs:boolean	Helmet	People
	fc:Accessory/fc:Mask/fc:Color/tt:ColorCluster	tt:ColorDescriptor	MaskColor	People
	fc:Accessory/fc:Mask/fc:Wear[text()]	xs:boolean	Mask	People
	fc:Accessory/fc:Opticals/fc:Wear[text()]	xs:boolean	Glasses	People
	fc:Age	tt:IntRange	AgeOnvif	People
	fc:FacialHair/fc:Mustache[text()]	xs:boolean	Mustache	People
	fc:FacialHair/fc:Sideburn[text()]	xs:boolean	Sideburn	People
	fc:FacialHair/fcBeard[text()]	xs:boolean	Beard	People
	fc:Gender[text()]	fc:Gender	Gender	People
	fc:Hair/fc:Color/tt:ColorCluster	tt:ColorDescriptor	HairColor	People
	fc:Hair/fc:Length[text()]	fc:Length	HairLength	People
	tt:Appearance/tt:Class/			
	tt:ClassCandidate/tt:Type[text()!='Face']	tt:ClassType	Class	
	tt:ClassCandidate/tt:Type[text()='Face']	tt:ClassType	Face	People
	tt:Type[text()!='HumanFace']	tt:ObjectType	Class	
	tt:Type[text()]	tt:ObjectType	VehicleType	Vehicles
	tt:Type[text()='HumanFace']	tt:ObjectType	Face	People
	tt:Appearance/tt:Color/tt:ColorCluster	tt:ColorDescriptor	VehicleColor	Vehicles
	tt:Appearance/tt:GeoLocation/			
	@lat	xs:double	Latitude	Location
	@lon	xs:double	Longitude	Location
	tt:Appearance/tt:LicensePlateInfo/			
	tt:CountryCode[text()]	xs:string	LicensePlateCountryCode	Vehicles
	tt:IssuingEntity[text()]	xs:string	LicensePlateIssuingEntity	Vehicles
	tt:PlateNumber[text()]	xs:string	LicensePlate	Vehicles
	tt:PlateType[text()]	tt:PlateType	LicensePlateType	Vehicles
	tt:Appearance/tt:VehicleInfo/			
	tt:Brand[text()]	xs:string	VehicleBrand	Vehicles
	tt:Model[text()]	xs:string	VehicleModel	Vehicles
	tt:Type[text()]	tt:VehicleType	VehicleType	Vehicles
	tt:Behaviour/tt:Speed[text()]	xs:float [m/s]	Speed	Vehicles
	tt:Extension/Properties/			

Root	Element or attribute XPath	Value type	Search property	Search category
	`Property[@name='Age']`	`xs:int` [y]	Age	People
	`Property[@name='Class']`		Class	
	`Property[@name='Color']`		VehicleColor	Vehicles
	`Property[@name='Gender']`	`fc:Gender`	Gender	People
	`Property[@name='Height']`	`xs:decimal` [cm]	Height	People
	`Property[@name='LicensePlate']`	`xs:string`	LicensePlate	Vehicles
`/tt:MetadataStream/tt:Extension/NavigationalData/`				
	`Latitude[text()]`	`xs:decimal`	Latitude	Location
	`Longitude[text()]`	`xs:decimal`	Longitude	Location
`/tt:MetadataStream/tt:Event/wsnt:NotificationMessage/ wsnt:Topic[ text()='tnsaxis:CameraApplicationPlatform/ALPV.AllPlates' ]/../wsnt:Message/tt:Message/tt:Data/`				
	`tt:SimpleItem[@Name='text']/@Value`	`xs:string`	LicensePlate	Vehicles

Labels used for ONVIF string enumerations

In general, the labels used for ONVIF string enumerations in the Smart Client search user interface correspond to the definitions in the ONVIF schema. For a few value types, the enumeration values in the ONVIF schema are unclear, and the label in the search user interface differs somewhat.

Search properties	Value type	ONVIF enumeration	Label (en-US)	Comment
ClothingBottomStyle	`bd:BottomsStyle`			
		`FornalPants`	Dress pants	
		`Jeans`	Jeans	
		`Other`	Other	
ClothingBottomCategory	`bd:BottomsCategory`			
		`Trousers`	Pants	
		`Shorts`	Shorts	
		`Skirt`	Skirt	
		`Other`	Other	
ClothingBottomGrain ClothingTopGrain	`bd:Grain`			
		`Plaid`	Plaid	
		`Decal`	Printed design	
		`PureColour`	Solid color	
		`Stria`	Striped	
		`Other`	Other	
BagCategory	`bd:KnapsackCategory`			

Search properties	Value type	ONVIF enumeration	Label (en-US)	Comment
		`Backpack`	Backpack	
		`SingleShoulderBag`	Single shoulder bag	
		`Other`	Other	
ShoesCategory	`bd:ShoesCategory`			
		`LeatherShoes`	Leather shoes	
		`Sandal`	Sandals	
		`Slipper`	Slippers	
		`Sneakers`	Sneakers	
		`Other`	Other	
ClothingTopCategory	`bd:TopsCategory`			
		`LongSleeve`	Long sleeves	
		`ShortSleeve`	Short sleeves	
		`Other`	Other	
ClothingTopStyle	`bd:TopsStyle`			
		`Dress`	Dress	
		`Jacket`	Jacket	
		`Overcoat`	Overcoat	
		`Sweater`	Sweater	
		`Tailor`	Tailor	
		`Other`	Other	
HairLength	`fc:Length`			
		`Long`	Long	
		`Medium`	Medium	
		`Short`	Short	
LicensePlateType	`tt:PlateType`			
		`Diplomat`	Diplomatic	
		`Police`	Police	
		`Normal`	Standard	
		`Temporary`	Temporary	
VehicleType	`tt:VehicleType`			
		`Bus`	Bus	
		`Car`	Car	
		`Truck`	Truck	
		`Bicycle`	Bicycle	

Search properties	Value type	ONVIF enumeration	Label (en-US)	Comment
		`Motorcycle`	Motorcycle	
		`Bike`	Bike	Extension: Added if `tt:Appearance/tt:Class/tt:Type` matches `Bike`
Class	`tt:ObjectType`			
		~~`Animal`~~		Not recognized
		~~`Barcode`~~		Not recognized
		`Bicycle`		
		`Bike`		Extension: Also counts as a `VehicleType`
		`Human`		Extension: Aliases added for `Human`: `Person`, `People`
		`HumanFace`		
		`LicensePlate`		
		`Vehicle`		
Class	`tt:ClassType`			
		~~`Animal`~~		Not recognized
		`Face`		
		`Human`		
		`Vehical`		
		~~`Other`~~		Not recognized

Search properties and categories

Properties that are based on extensions or alternatives to ONVIF are described in Extensions and alternatives to the ONVIF Analytics Service Specification.

To make metadata from an object available in Metadata search categories, both class values and search properties are considered. Currently, the algorithm is as follows:

- If no class value is available for an object, all class specific object properties are used to to infer search categories. For example, if no class value but both `Age` and `Speed` are available for an object, you can get hits for this object in both the People and the Vehicles search category.
- If any recognized class values are available for an object, you can get hits for this object in the corresponding search categories, and properties not related to those object classes are not considered. For example, if `Class="Vehical"` (and no other class values) and `Gender="Male"` are available for an object, you can only get hits for this object in the `Vehicles` search category.
- If an object has only unrecognized class values, properties are not considered, and you will not get hits for this object in any search category.

The properties `Latitude` and `Longitude` are not related to any specific object class.

> As a consequence, if any object class value (even unrecognized) is available, you cannot get hits for this object in the `Location` search category.

But, for the purpose of Metadata search, the `/tt:MetadataStream` element itself is also considered an object, and you can get hits in the `Location` search category if `/tt:MetadataStream/tt:Extension/NavigationalData/Latitude` or `/tt:MetadataStream/tt:Extension/NavigationalData/Longitude` are available.

Object color

XProtect supports two representations of object color:

- The ONVIF Color descriptor (ONVIF Analytics Service Specification, 5.3.4 Color descriptor)
  The colorspace (`YCbCr` or `RGB`) can be set on frame, object, and `ColorCluster` level. Default is `YCbCr` .
- An XProtect extension: HTML (RGB) hex codes or HTML/X11 color names
  Described further in Color extension.

Internally, color property values are processed as `System.Drawing.Color` , i.e. RGB, values.

In the Smart Client, color filters are available for a number of properties:

- Clothing and belongings (5.3.14 Human body descriptor):
  - Clothing Bottoms Color
  - Clothing Tops Color
  - Gloves Color
  - Scarf Color
  - Shoes Color
  - Bag Color
- Face (5.3.13 Face descriptor):
  - Hair Color
  - Hat Color
  - Helmet Color
  - Mask Color
- Vehicles:
  - Object Color (5.3.4 Color descriptor)

In the filters, color values are mapped to closest RGB color in a list specific for each search property. The color names and values are as defined in `System.Windows.Media.Color` with a few changes.

Name	Hex	Clothing	Vehicles	Hair
`Beige`	`#F5F5DC`	X	X	
`Black`	`#000000`	X	X	X
`Blond`	`#B88B50`			X
`Blue`	`#0000FF`	X	X	
`Brown`	`#996633`	X		X
`Gray`	`#808080`	X	X	X
`Green`	`#008000`	X	X	
`Orange`	`#FFA500`	X		
`Pink`	`#FFC0CB`	X		
`Purple`	`#800080`	X		
`Red`	`#FF0000`	X	X	X
`White`	`#FFFFFF`	X	X	X
`Yellow`	`#FFFF00`	X	X	

Compared to `System.Windows.Media.Color` , there are the following changes:

- `Blond` added
- `Brown` value changed from `#A52A2A` to `#996633` .

The color metric is based on the **weighted Euclidean distance** algorithm from Color difference or Colour metric.

Extensions and alternatives to the ONVIF Analytics Service Specification

Compared to the ONVIF Analytics Service Specification, XProtect Metadata search extends or supports alternatives for several object descriptors.

ONVIF object descriptor	Search properties
Class	Class
Color	VehicleColor
LicensePlate	LicensePlate
HumanFace and HumanBody	Face, Age, Gender, and Height
GeoLocation	Latitude, Longitude

Each extension is described in the following sections.

Class extension

XProtect extends the ONVIF Class descriptor as follows:

- Any `tt:Class/tt:ClassCandidate/tt:Type` value is allowed (in addition to the `tt:ClassType` enumeration in the ONVIF Analytics Service Specification)
  `tt:Class/tt:ClassCandidate/tt:Likelihood` is ignored
- Any `tt:Appearance/tt:Class/tt:Type` value is allowed (in addition to the `tt:ObjectType` enumeration in the ONVIF Analytics Service Specification)
  `tt:Class/tt:Likelihood` is ignored
- `tt:Extension/Properties/Property[@name = "Class"]` also provides Class type values

All of `tt:Class/tt:Type`, `tt:Class/tt:ClassCandidate/tt:Type`, and `tt:Extension/Properties/Property[@name = "Class"]` are considered equally, that is, an object could be classified simultaneously as both `Human`, `Bike`, and `LicensePlate`, and would appear in both People and Vehicles searches.

The following Class type values are recognized for Metadata search:

Class	Search category
`Face`, `HumanFace`	People
`Human`, `Person`, `People`	People
`Vehical`, `Vehicle`, `Bicycle`, `Bike`	Vehicles
`LicensePlate`	Vehicles

A few `tt:ClassType` and `tt:ObjectType` values from the ONVIF Analytics Service Specification are ignored as there isn't a corresponding search category:

- `Animal`
- `Barcode`
- `Other`

`Person` and `People` are considered aliases for `Human`.

The occurrence of a `Face` or `HumanFace` object also maps to the `Face` search property of the People search category.

The occurrence of a `Bike` object also maps to the value of the `VehicleType` search property of the Vehicles search category.

Color extension

In addition to the ONVIF color representation, XProtect uses an extension to support a simpler representation of object color:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="13">
        ...
        <tt:Extension>
          <Properties>
            <Property name="Color">#564565</Property>
          </Properties>
        </tt:Extension>
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

The color format and values are those recognized by .NET `System.Drawing.ColorTranslator.FromHtml()`:

- "#rgb": 3 hex digits (red, green, blue), for example `#546`.
- "#rrggbb": 6 hex digits (red, green, blue), for example `#564565`.
- .NET `System.Drawing.KnownColor` names, based on CSS/X11 named colors. While the CSS named colors include "grey" aliases for "gray" colors, only the 'gray' names are supported by `System.Drawing.ColorTranslator.FromHtml()`.

`<Property name="Color">` is mapped to the search property `VehicleColor`, related to the search category `Vehicles`.

LicensePlate extension

Three different XPaths contribute `LicensePlate` values:

- The ONVIF `tt:LicensePlateInfo` element
- An XProtect extension
- An Axis extension (Axis ALPV)

ONVIF license plate example:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="13">
        <tt:Appearance>
          ...
          <tt:LicensePlateInfo>
            <tt:PlateNumber>AB12345</tt:PlateNumber>
          </tt:LicensePlateInfo>
        </tt:Appearance>
        ...
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

XProtect license plate example:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="13">
        ...
        <tt:Extension>
          <Properties>
            ...
            <Property name="LicensePlate">AB12345</Property>
          </Properties>
        </tt:Extension>
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

Axis ALPV license plate example:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:Event>
    <wsnt:NotificationMessage
      xmlns:tns1="http://www.onvif.org/ver10/topics"
      xmlns:tnsaxis="http://www.axis.com/2009/event/topics"
      xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
      xmlns:wsa5="http://www.w3.org/2005/08/addressing">
      <wsnt:Topic Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">tnsaxis:CameraApplicationPlatform/ALPV.
      <wsnt:ProducerReference>
        <wsa5:Address>uri://face2652-97c8-4d13-a4ce-723baa9d6c2c/ProducerReference</wsa5:Address>
      </wsnt:ProducerReference>
      <wsnt:Message>
        <tt:Message UtcTime="2019-11-29T09:51:08.524672Z">
          <tt:Source/>
          <tt:Data>
            ...
            <tt:SimpleItem Name="text" Value="AB12345"/>
            ...
          </tt:Data>
        </tt:Message>
      </wsnt:Message>
    </wsnt:NotificationMessage>
  </tt:Event>
</tt:MetadataStream>
```

HumanFace and HumanBody extension

In addition to the ONVIF `HumanFace` and `HumanBody` object descriptors, XProtect supports properties for describing a human object through an extension.

Also, in addition to the ONVIF object class `Human`, XProtect recognizes `Person` and `People` as aliases for `Human`.

Human properties example:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="12">
          ...
          <Properties>
            <Property name="Gender">Male</Property>
            <Property name="Height">1.81</Property>
            <Property name="Age">48</Property>
          </Properties>
        </tt:Extension>
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

GeoLocation extension

In addition to the object location introduced in the ONVIF 18.06 release,

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="13">
        <tt:Appearance>
          ..
          <tt:GeoLocation lat="16.0735273296356" lon="-27.2426441904356" />
        </tt:Appearance>
        ...
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

XProtect has supported navigational data related to the metadata stream (i.e., the camera itself) since 2014:

```
<?xml version="1.0" encoding="utf-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
    ...
  </tt:VideoAnalytics>
  <tt:Extension>
    <NavigationalData version="1.0">
      <Latitude>52.069926</Latitude>
      <Longitude>11.796875</Longitude>
      ...
    </NavigationalData>
  </tt:Extension>
</tt:MetadataStream>
```

For Metadata search, the location values are expected to use same notation and geodetic reference as ONVIF: decimal degrees, WGS84.

To support backwards and forwards compatibility, ONVIF applies a schema version and extension policy. For a number of MetadataStream extension points, the ONVIF schema requires vendor element extensions to be declared in a separate vendor specific namespace. `/tt:MetadataStream/tt:Extension` is one of the extension points in which content elements is now required to be in a vendor namespace. But, unless you enforce schema validation, this is very unlikely to cause problems.

Examples

A few complete examples illustrate the use of ONVIF metadata for Metadata search.

Vehicle example

This example includes the use of these extensions:

- A non-spec class type value accepted in `tt:Class/tt:ClassCandidate/tt:Type`; `tt:Class/tt:Extension/tt:OtherTypes` is not used
- A simple representation of object color

```
<?xml version="1.0"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="13">
        <tt:Appearance>
          <tt:Shape><!-- Not used for Metadata search  -->
            <tt:BoundingBox bottom="20.0" top="30.0" right="100.0" left="80.0" />
            <tt:CenterOfGravity x="60.0" y="50.0" />
          </tt:Shape>
          <tt:Class>
            <tt:ClassCandidate>
              <tt:Type>Vehicle</tt:Type><!-- 'Vehicle' accepted as an alias for 'Vehical' -->
              <tt:Likelihood>1.0</tt:Likelihood>
            </tt:ClassCandidate>
          </tt:Class>
          <tt:GeoLocation lat="16.0735273296356" lon="-27.2426441904356" />
          <tt:VehicleInfo>
            <tt:Type>Car</tt:Type>
          </tt:VehicleInfo>
          <tt:LicensePlateInfo>
            <tt:PlateNumber>AB12345</tt:PlateNumber>
          </tt:LicensePlateInfo>
        </tt:Appearance>
        <tt:Behaviour>
          <tt:Speed>56</tt:Speed>
        </tt:Behaviour>
        <tt:Extension>
          <Properties>
            <Property name="Color">#564565</Property><!-- Object color extension -->
          </Properties>
        </tt:Extension>
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
</tt:MetadataStream>
```

Human and Face example

This example includes the use of these extensions:

- `tt:Likelihood` is ignored, so the sum does not need to be equal to or less than 1.0
- A simple representation of human properties
- Geo location related to the camera

```
<?xml version="1.0"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
  <tt:VideoAnalytics>
    <tt:Frame UtcTime="2018-01-02T09:28:09.813488Z">
      <tt:Object ObjectId="12">
        <tt:Appearance>
          <tt:Shape><!-- Not used for Metadata search  -->
            <tt:BoundingBox bottom="20.0" top="30.0" right="100.0" left="80.0"/>
            <tt:CenterOfGravity x="60.0" y="50.0"/>
          </tt:Shape>
          <tt:Class>
            <tt:ClassCandidate>
              <tt:Type>Human</tt:Type>
              <tt:Likelihood>1.0</tt:Likelihood><!-- tt:Likelihood is ignored -->
            </tt:ClassCandidate>
            <tt:ClassCandidate>
              <tt:Type>Face</tt:Type>
              <tt:Likelihood>1.0</tt:Likelihood>
            </tt:ClassCandidate>
          </tt:Class>
        </tt:Appearance>
        <tt:Extension>
          <Properties>
            <Property name="Gender">Male</Property><!-- Human property extension -->
            <Property name="Height">1.81</Property><!-- Human property extension -->
            <Property name="Age">48</Property><!-- Human property extension -->
          </Properties>
        </tt:Extension>
      </tt:Object>
    </tt:Frame>
  </tt:VideoAnalytics>
  <tt:Extension>
    <NavigationalData version="1.0"><!-- Navigational data extension -->
      <Latitude>16.0735273296356</Latitude>
      <Longitude>-27.2426441904356</Longitude>
    </NavigationalData>
</tt:MetadataStream>
```

Guidelines

Alternative representations of object class type are supported for Metadata search in order to support existing devices. For new development, adhere to the ONVIF Analytics Service Specification Version 23.12.

The ONVIF MetadataStream schema allows several `tt:Frame` elements in the same `tt:VideoAnalytics` element, and several `tt:VideoAnalystics` elements within a `tt:MetadataStream` element. We strongly recommend that you have only one `tt:Frame` and one `tt:VideoAnalytics` element per packet sent to the recording server. This makes it easier to find data based on timestamps and to associate metadata with the correct video frames.

Help and troubleshooting

For further help and troubleshooting, refer to the Milestone Developer Forum.

If you are a Milestone Technology Partner, you can also raise the issue through the Milestone support case system.

References

- ONVIF Network Interface Specifications
- ONVIF Analytics Service Specification Version 23.12
- ONVIF XML Schema Version and Extension Handling White Paper
- W3C CSS Color Module Level 3

- .NET System.Drawing.KnownColor Enum
- Introduction to Metadata
- Milestone Developer Forum

About Milestone Systems

Milestone Systems is a leading provider of data-driven video technology software in and beyond security that helps the world see how to ensure safety, protect assets, and increase business efficiency. Milestone enables an open platform community that drives collaboration and innovation in the development and use of network video technology, with reliable and scalable solutions that are proven in more than 500,000 customer sites worldwide. Founded in 1998, Milestone is a stand-alone company in the Canon Group.

www.milestonesystems.com