Contents

Trigger an alarm externally

This is a concept solution that creates an alarm in the Smart Client when something happens in the realm monitored by XProtect Milestone. The solution includes both documentation and sample code.

Introducing the task

The task is to direct the user to video from a specific camera in the XProtect Smart Client triggered by your standalone program.

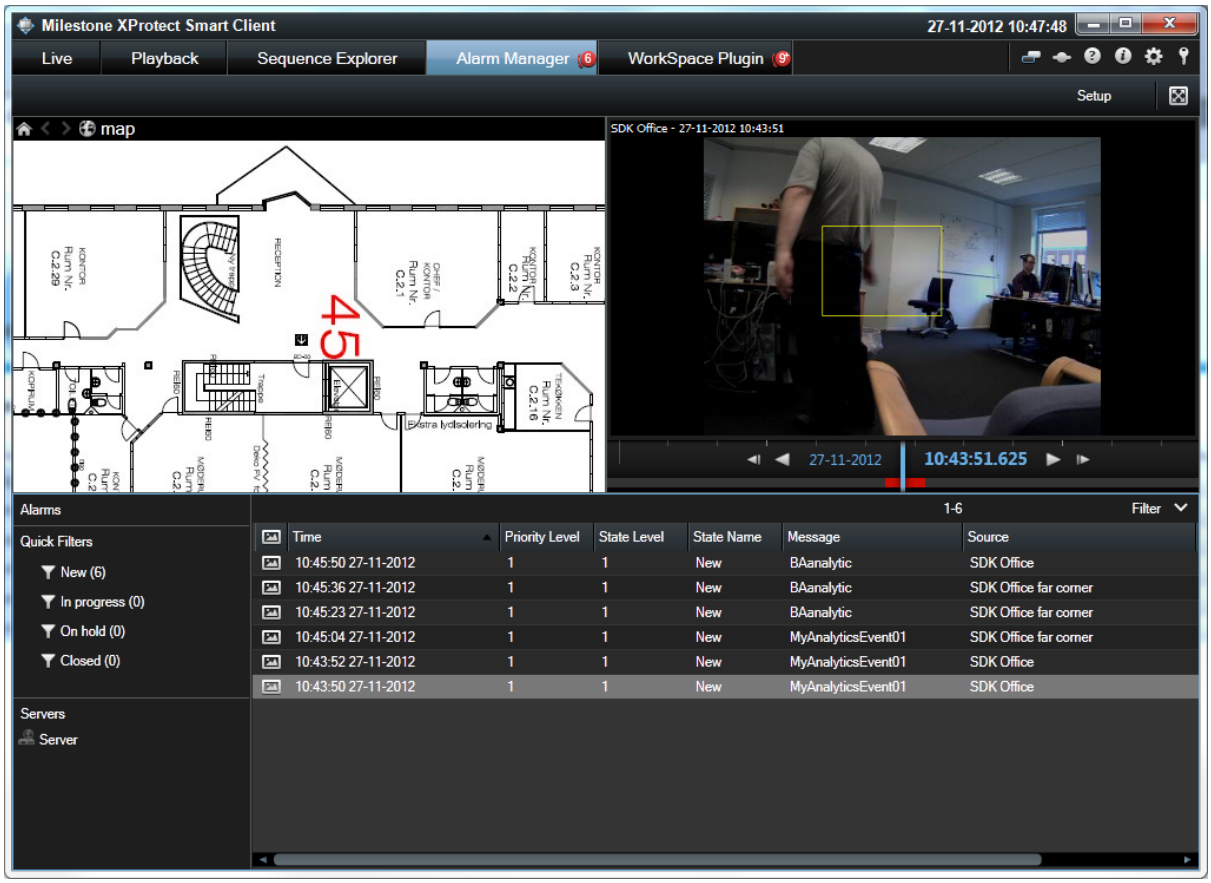

Figure 1: Smart Client Alarm Manager.

Illustration of how a user can see "alarms". Note how the user gets a list of alarms in the Smart Client, when picking an alarm the users gets a picture from the camera that is the source of the event recorded at the time of alarm.

The way to achieve this is to tell the Milestone XProtect Event Server that you are triggering an event.

Figure 2: Simplified dataflow

Your standalone program is responsible for telling the Event Server that there is an event. Your standalone program might be using camera data, video data or data from other sensor devices in the same rooms as a camera.

Introducing the samples

The scope for this solution is to describe the simplest way of triggering an alarm. You should be able to take the code samples, follow the instructions in this document and have the solution up and running within a few minutes.

The solutions describe "the simplest way" and are at the same time the recommended way of doing things.

There are 2 samples, showing 2 methods.

The first sample is *Trigger Analytics Event with SDK*, this sample is using the Milestone SDK .NET Library functions, which is the recommended method to use.

The second sample is *Trigger Analytics Event with XML* this sample is using SOAP and should be easy to implement the widest possible range of environments and development tools.

All sample code is written in C# even though the *Trigger Analytics Event with XML* sample is targeted to programmers in a non-Windows environment like Linux.

Definitions, Acronyms and Abbreviations

Name	Description
XPE	XProtect Enterprise
XPCO	XProtect Corporate
SC	XProtect Smart Client
SOAP	SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format.

Overview

This solution consists of a standalone program that triggers an event in a Milestone product which in turn creates an alarm that pops up in the Smart Client.

There are basically three parts in this solution: Configuration, Triggering and Presentation, see Figure 3: A logical overview of the solution.

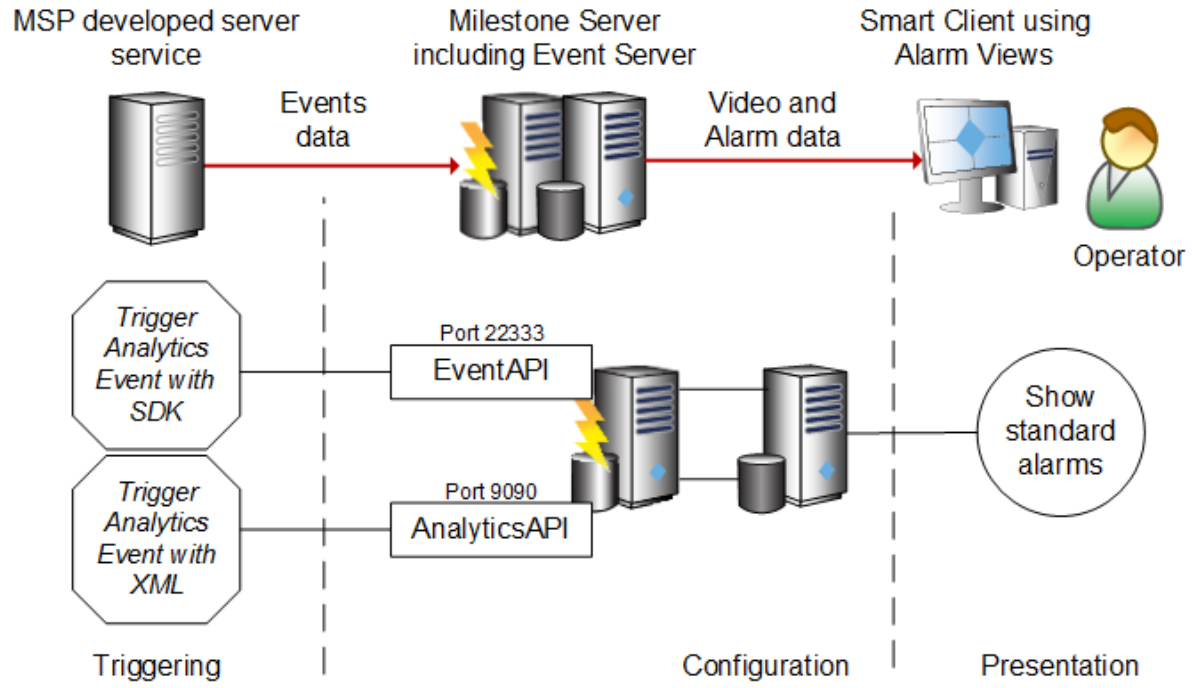

Figure 3: A logical overview of the solution.

The triggering part consists of two parallel flows: XML and SDK:

- Triggering the event with help of the MIP SDK illustrates the use of the MIP SDK. By using the SDK you will get a lot of functionality that you don't have to implement yourself. Also, your programming interface will be on a level "above" the XML so you don't have to bother with that. And possible future changes like the protocol are handled by the SDK.
- Triggering the event with help of an XML document is to be used when the triggering program executes in a non-Windows environment, or when the MIP SDK cannot be used for some reason. The strength with this sample is that almost any program or device can integrate with the Milestone system.

The configuration is described in a step by step way for both Enterprise and Corporate. The fundamentals of the configuration are that you create an analytics event and an alarm definition and the analytics event triggers the alarm. An important fact is also that the alarm source is "All Cameras" in the alarm definition. And to make the source specific, the analytics event will be created with a specific camera. This gives the possibility to set the source programmatically for each event triggered.

The presentation of the alarm is in the Smart Client. The standard alarm list is used so no configuration or programming has to be done here. There will be a brief description in this document of the data fields used in the code sample.

The minimal solution

Configuration of the Milestone product

There are two parts in the configuration: creating an "Analytics Event" and creating an "Alarm Definition". The configuration steps are more or less the same in Enterprise and Corporate but they are described separately anyway for clarity.

Enterprise configuration

1. Open the Enterprise Management Application.
2. Click the menu *Application Settings* and then *Application Behavior*.
3. Click the *Analytics Events Settings* in the left side.
4. Make sure that analytics events are enabled.
5. Click OK.
6. Expand the *Advanced Configuration* node.
7. Expand *Events and Output* node.
8. Right-click *Analytics Events* node and select *Create New*.
9. Give the event a name, e.g. MyAnalyticsEvent01.
10. Click OK.
11. Expand the *Alarms* node under *Advanced Configuration*.
12. Right-click *Alarm Definitions* and select *Create New*.
13. Give the alarm definition a name, e.g. MyAnalyticsAlarm01.
14. Select *Analytics Event* in the first drop down under *Triggering event* and then your event (MyAnalyticsEvent01) in the second drop down.
15. Select All Cameras as *Sources*. There must be a source to the alarm and if the alarm is configured to have *All Cameras* as source, a specific camera can be set programmatically when the event is triggered.
16. Click OK.
17. Click *Save and Apply Configuration* on the toolbar.

Corporate configuration

1. Open the Corporate Management Client.
2. Click the *Analytics Events* node under *Rules and Events* top node in the left control.
3. Right-click the *Analytics Event* node in the middle control and select *Add New*.
4. Give the event a name, e.g. MyAnalyticsEvent01.
5. Click Save on the toolbar.
6. Expand the *Alarms* top node in the left control and click *Alarm Definitions*.
7. Right-click the *Alarm Definitions* node in the middle control and select *Add New*.
8. Give the alarm definition a name, e.g. MyAnalyticsAlarm01.
9. Select *Analytics Event* in the first drop down under *Triggering event* and you're your event (MyAnalyticsEvent01) in the second drop down.

10. Select All Cameras as *Sources*. There must be a source to the alarm and if the alarm is configured to have *All Cameras* as source, a specific camera can be set programmatically when the event is triggered.
11. Click Save on the toolbar.

Triggering the alarm

The event is triggered by a standalone program written by a partner. There are two samples to illustrate this: one program that uses the MIP SDK and one that does not.

The program that does not use the MIP SDK is communicating with the Event Server with an analytics XML document sent over TCP is called *Trigger Analytics Events with XML*, see the separate section below.

The program that use the MIP SDK takes full advantage of the MIP SDK library is called *Trigger Analytics Event with SDK*, see the separate section below.

Both samples create the same event data except for the event source. In the XML document there is the possibility of specifying the IP address of the camera as the event source. The Event Server will then lookup the camera defined on this address. The IP address cannot be used when creating the event from the SDK but here you have the possibility to search for the camera directly in the configuration. You can also specify the GUID as the camera source, see the GUID comment in the *AnalyticsEventOverlay.xml* file in the *TriggerAnalyticsEventXML* project.

Also, the tag (the CustomTag element) is different in the two samples so we can see which alarm comes from which program.

Trigger Analytics Event with SDK code sample

This sample triggers the same event as in the XML sample but using the MIP SDK. It sends exactly the same data as in the XML sample (apart from the comments in the common section above).

The code that actually triggers the event is less than 30 lines of code, see figure 3.

```
EventSource eventSource = new EventSource()
{
              FQID = _selectedItem.FQID,
              Name =  selectedItem.Name
};
EventHeader eventHeader = new EventHeader()
{
              ID = Guid.NewGuid(),
              Type = "MyType",
              Timestamp = DateTime.Now,
              Message = "MyAnalyticsEvent01",
              Source = eventSource,
              CustomTag = "TagFromC#"
};

AnalyticsEvent analyticsEvent = new AnalyticsEvent();
analyticsEvent.EventHeader = eventHeader;
analyticsEvent.Location = "Event location 1";
analyticsEvent.Description = "Analytics event description.";
analyticsEvent.Vendor = new Vendor();
analyticsEvent.Vendor.Name = "My Smart Video";

analyticsEvent.ObjectList = new AnalyticsObjectList();
analyticsEvent.ObjectList.Add(GetRectangle());
```

```
EnvironmentManager.Instance.SendMessage(
              new VideoOS.Platform.Messaging.Message(MessageId.Server.NewEventCommand)
                  { Data = analyticsEvent });
```

Figure 4: Trigger the event with the SDK.

This sample includes a “picker form”, used interactively to select a camera for the event source. This is however not the case in real life. In a real life scenario the camera would come from a pre-defined camera or a non-interactive lookup of the relevant camera.

Further documentation on the AnalyticsEvent class is found in the MIP SDK Documentation.

Trigger Analytics Events with XML code sample

This sample is written in C# but the purpose is to illustrate a program on another platform. One real life scenario could be C++ on Linux. The code is using the *Socket* class to send an XML document to the event server, see figure 2 for the XML. The reason for using the Socket class is that there are similar classes in other languages so “Copy – Paste” from this sample code should be fairly straightforward.

```
<?xml version="1.0" encoding="utf-8"?>
<AnalyticsEvent xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
                xmlns="urn:milestone-systems">
  <EventHeader>
    <ID>00000000-0000-0000-0000-000000000000</ID>
    <Timestamp>$timestamp$</Timestamp>

    <Type>MyType</Type>

    <!-- Insert Event Message here -->
    <Message>MyAnalyticsEvent01</Message>

    <CustomTag>TagFromXML</CustomTag>

    <Source>

      <!-- Insert camera URI here, if you don't have the GUID. -->
      <!-- (For multichannel devices, URI may contain channel number after ',') -->
      <Name>10.10.17.2</Name>

    </Source>
  </EventHeader>

  <Description>
    Analytics event description.
  </Description>

  <Location>
    Event location 1
  </Location>

  <Vendor>
    <Name>My Smart Video</Name>
  </Vendor>

</AnalyticsEvent>
```

Figure 5: A simple analytics XML document.

Please note that the text value *$timestamp$* in the <Timestamp> tag is replaced by the sample code to a timestamp string in the format “yyyy-MM-ddTHH:mm:sszzz” before it is sent.

Further documentation on the AnalyticsEvent xml is found in the MIP SDK Documentation and the AnalyticsEvent.xsd which is part of the MIP SDK ("C:\Program Files (x86)\Milestone\MIPSDK\XSD\AnalyticsEvent.xsd").

Presentation of the alarm

After the analytics event has been triggered successfully, an alarm is created and will show up in the Smart Client, see *Figure 6* for a screenshot. This solution uses the alarm list in the Smart Client as it is. You can define which fields that is possible to show in the alarm list. They are defined in the *Alarm Data Settings* node under the *Alarms* top node. For example, the Smart Client operator can sort the alarms on the custom tag.

If you double click the alarm row you get the details and here you can see the data that was created by the triggering program, see *Figure 7: A screenshot of the alarm details*..

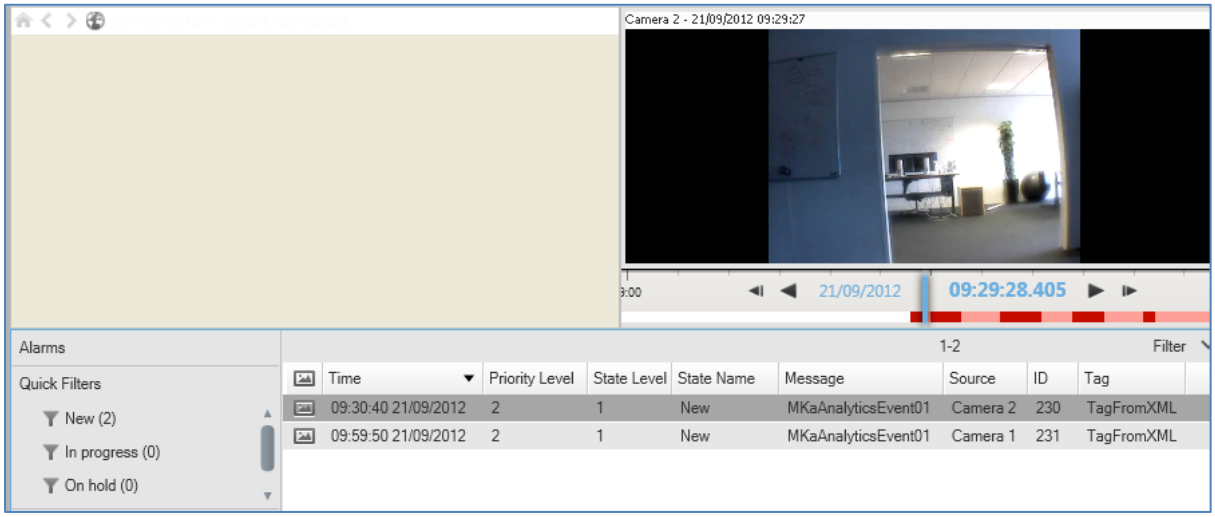

Figure 6: A screenshot of the alarm list in the Smart Client.

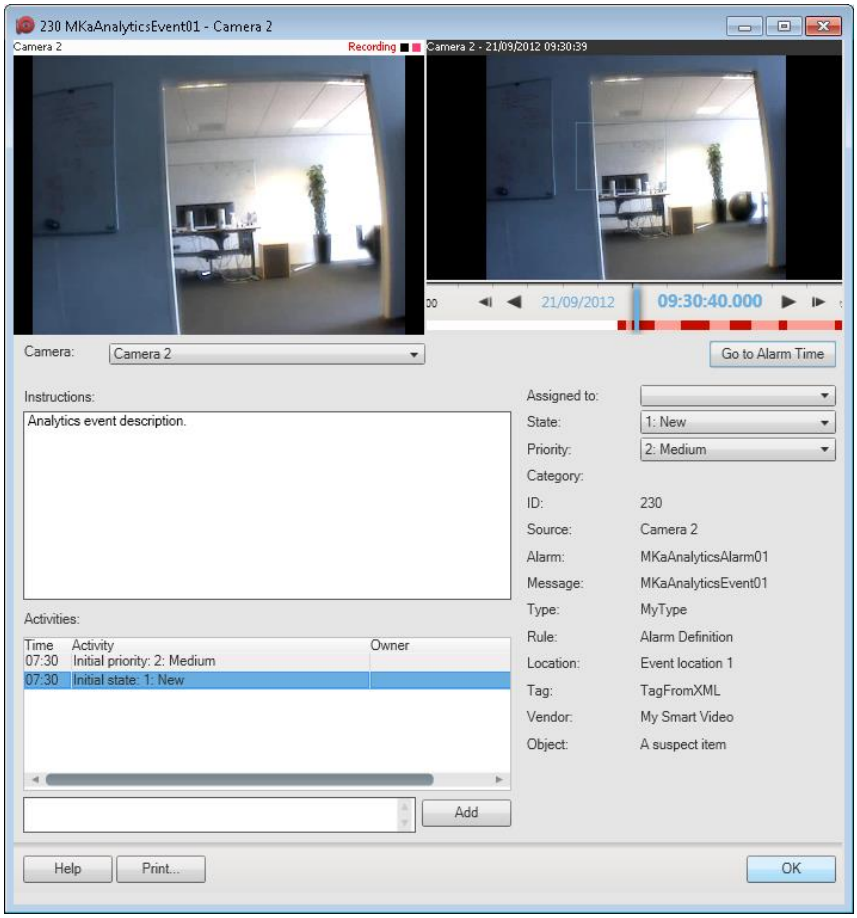

Figure 7: A screenshot of the alarm details.

In an advanced scenario a Smart Client plug-in can execute and modify the alarm view, for example show additional (attached) images.

Note:

The Smart Client Alarm Preview view (which is also part of the Alarm Manager work space) displays the camera images from the time of the alarm and puts the overlay on top of the images. The implementation is so that the drawn rectangle (or polygon) is drawn with pixel width 1.

Similarly the Smart Client Alarm Preview view displays the overlay in a short period, I think the period is 1 second before the event and 3 seconds after the event. It does not use the EndTime of the event for this.

The behaviour of the Smart Client Alarm Preview view is that when an Alarm is chosen in Alarm List that Alarm Preview starts playing back the camera footage from the time of event and forward, you will then see overlay only for less than 3 seconds.

Unfortunately this functionality is not configurable. You cannot configure a longer period of display.

How to run the code samples in your system

First, your system must be configured. Follow the steps in the *Configuration of the Milestone product* section.

Also, it is a prerequisite that you have at least one camera, let us say “Camera 1” that is set to record always.

Then you can build and run the two code samples provided.

Let us start with the *Trigger Analytics Events with SDK* sample.

1. Set the Text property of the *txtAnalyticsEventName* text box to your analytics event name (e.g. MyAnalyticsEvent01).
2. Set the *TriggerAnalyticsEventSDK* project as the “StartUp project” and run.
3. Log on to your system.
4. Click the *Select event source* button and select your “Camera 1”.
5. Click the Send analytics event button.
6. Open the Smart Client and verify that the recently triggered alarm is in the alarm list.

The next sample Trigger Analytics Event with XML is in the same solution and to run it you have to:

1. Open the TriggerAlarmFromExternal Visual Studio solution.
2. Open *AnalyticsEvent.xml* in the *TriggerAnalyticsEventXML* project.
3. Update the text in the <Message> element to your analytics event name (e.g. MyAnalyticsEvent01).
4. Update the <Name> element (inside <Source>) to the IP address of your “Camera 1”.
5. Build the *TriggerAnalyticsEventXML* project.
6. Set the *TriggerAnalyticsEventXML* project as the “StartUp project” and run.
7. Click *Insert Analytics event XML* button.
8. Click the Send XML button.
9. The following text should now be in the Socket receive text box.

```
HTTP/1.1 200 OK
Connection: Close
Content-Length: 0
```

10. Open the Smart Client and verify that the recently triggered alarm is in the alarm list.

The advanced solution - a teaser

Defining image overlays

In the two samples there are two options: to send overlay objects or not.

Show custom information in the Smart Client.

When you make send in analytics events to the Event Server you can include more information. Two text fields are especially interesting as they can be used in the Smart Client without doing any Smart Client development, but just using the Smart Client as it is.

The two fields in the Smart Client Alarm List are Tag and Object. In the Analytics object/XML they are CustomTag and ObjectList-Object-Value.

```
<CustomTag>TagFromXML</CustomTag>

<ObjectList>
  <Object>
    <Name>Rectangle</Name>
    <Value>A suspect item</Value>
  </Object>
</ObjectList>
```

Figure 8: Example XML

To make sure the Alarm List in Smart Client will show these fields you must do the following configuration in the Management Client / Management Application.

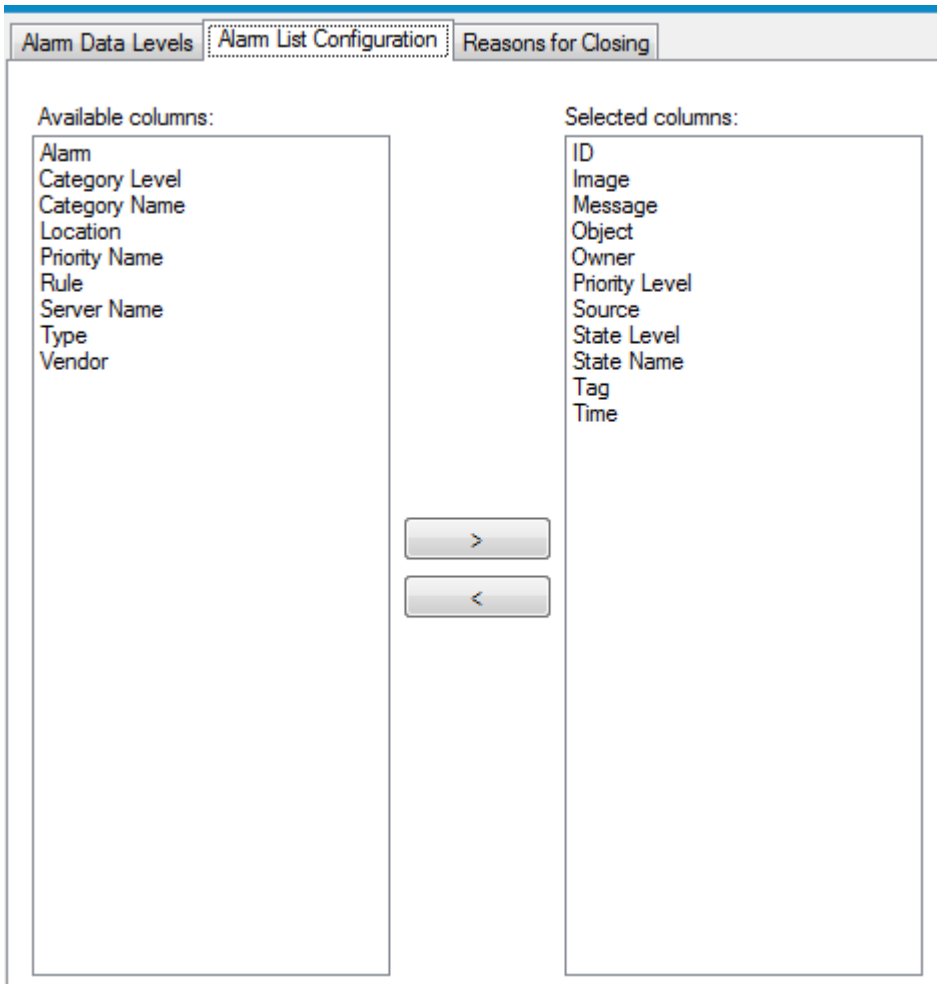

Figure 9: Configure Alarm List

You will get the following result in the Smart Client:

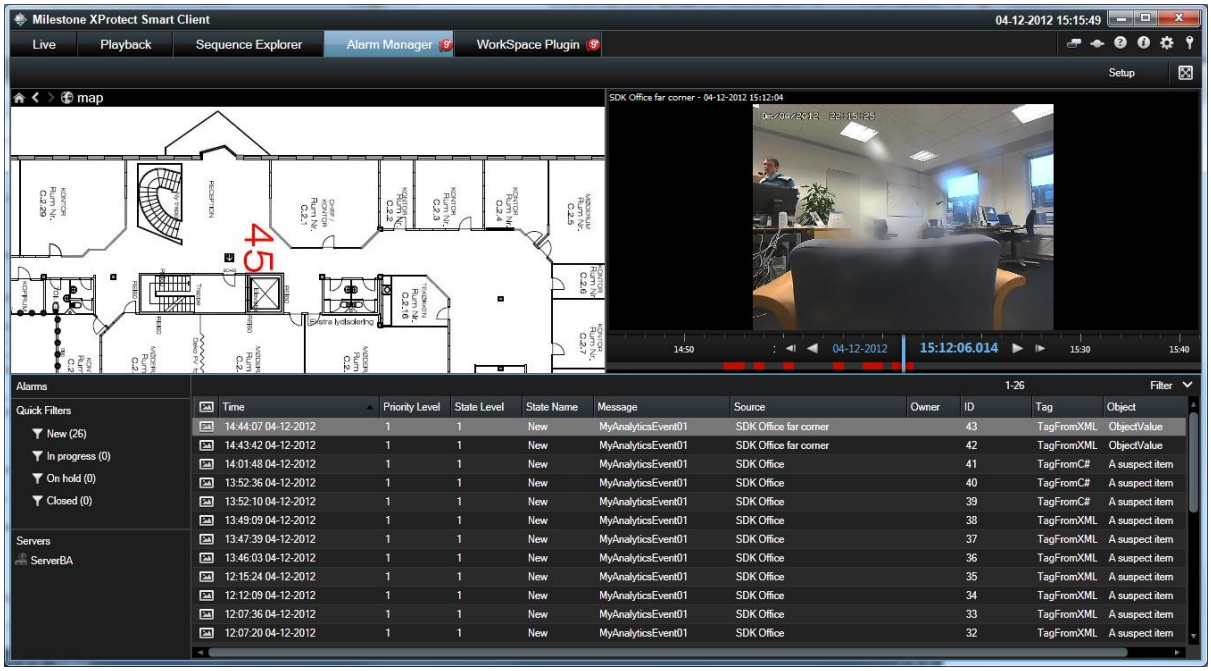

Figure 10: Smart Client Alarm List showing Tag and Object columns

The Smart Client does also have the ability to show an Event List, when you click Setup you will see

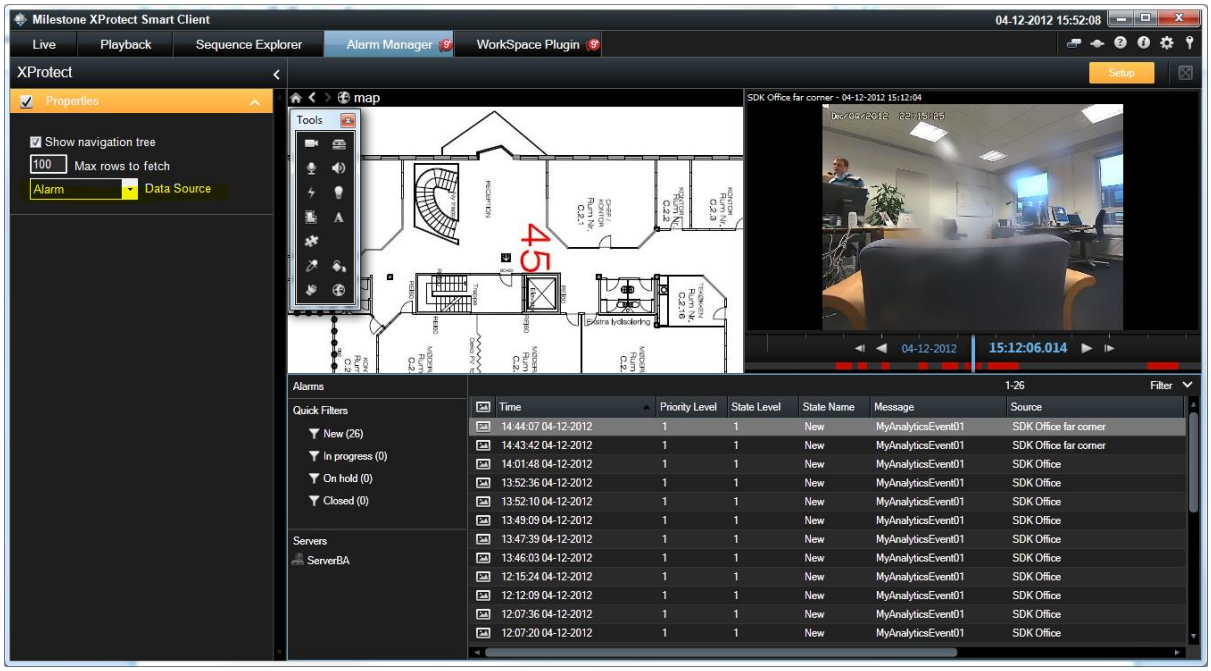

Figure 11: Smart Client setup to change from Alarm to Event in the "Alarm" List

You will see this instead

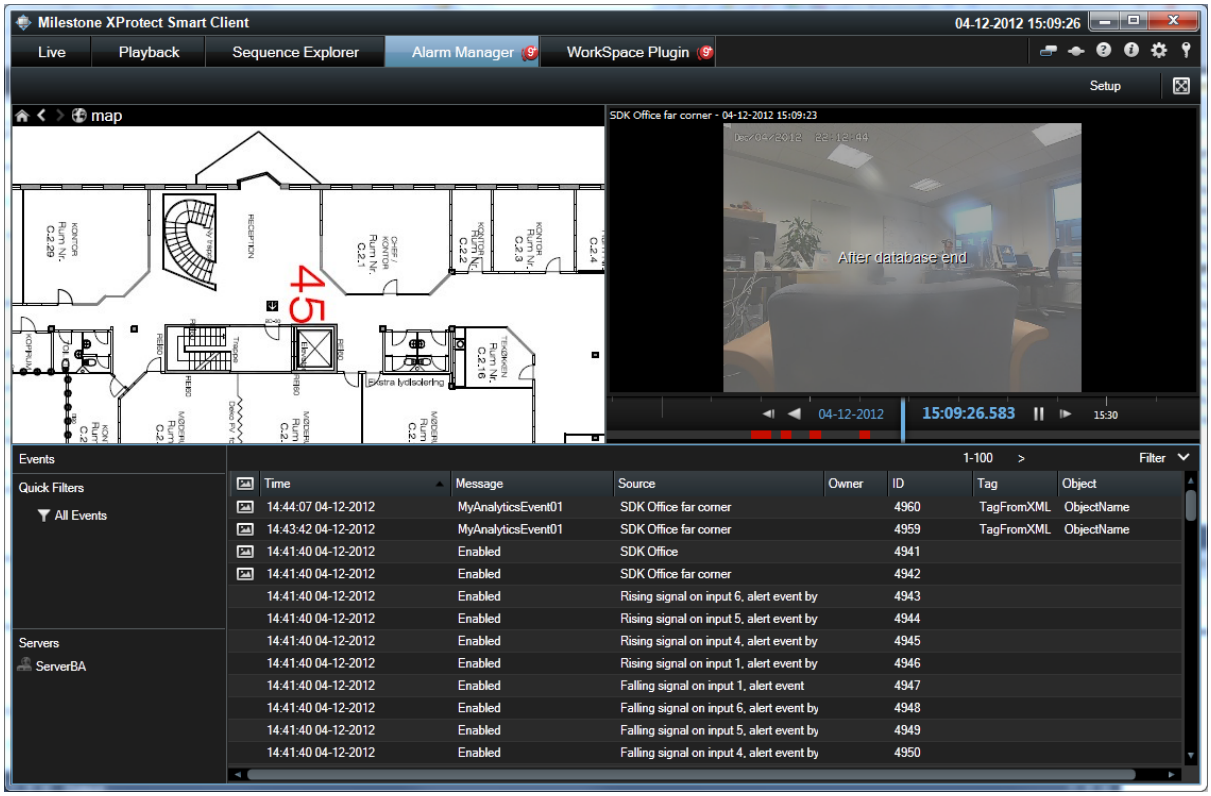

Figure 12: Smart Client Event List

Note: Some versions of Smart Client have the Event List using Object-Name while the Alarm List uses Object-Value. Future Smart Clients will use Object-Value in the object column. For best backwards compatibility you can put the same value in both Object-Value and Object-Name.

The correlation between Smart Client Alarm List and the Analytics Events tags.

The following table shows the Smart Client Alarm List columns and how they correspond with the Analytics Event. The columns entries marked in red are not shown in the Smart Client per default which means they must setup to be shown in Alarm List Configuration in the Manager (Management Client/Management Application), and afterwards must be setup to be shown in the Smart Client Alarm list (right-click the Alarm List header row and pick the columns to include).

Smart Client Alarm list column	Inherit from Analytics Event	Alarm Management feature	Analytics Event xml tag	Analytics Event class member	Notes
Image					
Time	Yes		<Timestamp>x </Timestamp>	.EventHeader .Timestamp	
Priority Level		Yes			
State Level		Yes			
State Name		Yes			

Message	Yes		<Message>x </Message>	.EventHeader .Message	Name of the Analytics Event as the setup in the Manager
Source	Yes		<Source>… </Source>	.EventHeader .Source	Source device (most often camera)
Owner		Yes			
ID					
Priority Name		Yes			
Category Level		Yes			
Category Name		Yes			
Alarm					Name of the Alarm Definition as setup in the Manager
Rule			<RuleList> <Rule> <Type>x </Type> </Rule> </RuleList>		If left out "Alarm Definition" is displayed, otherwise [blank] is displayed
Vendor	Yes		<Vendor> <Name>x </Name> </Vendor>	.Vendor.Name	
Location	Yes		<Location>x </Location>	.Location	
Tag	Yes		<CustomTag>x </CustomTag>	.EventHeader .CustomTag	
Type	Yes		<Type>x </Type>	.EventHeader .Type	
Object	Yes		<ObjectList> <Object> <Value>x </Value> </Object> </ObjectList>	.ObjectList[0] .Value	If <ObjectList> contain more than one <Object> only the valuet of the first object is displayed.
Server Name	(Yes)				

You can build your own Smart Client Plug-in to act as a replacement for the native Smart Client Alarm List, you cannot display more of the information from the Analytics Event in the native Smart Client Alarm List than what is listed.

Object, Tag, Type and Location are the ones you as a developer can utilize freely and will be able to display in the Smart Client Alarm List.

Custom action in the Smart Client

You do not have to do anything extra to make the alarms appear in the Smart Client alarm list. You can however customise which fields to be shown in the alarm list.

If you want additional features in the Smart Client you can create a background plug-in that listens to new alarm notifications. This plug-in can filter out the interesting alarms and access all data defined in the event / alarm and do whatever action it likes.