



VideoXpert Event Monitor Service

Contents

- Overview..... 3
- Installation..... 3
 - Prerequisites 3
 - Versioning And Licensing Information 3
 - Procedure 3
- Default XML Files..... 3
- VideoXpert Event Monitor Configuration..... 3
- Camera Configuration..... 4
- Monitor Configuration..... 4
- Supported Alarm Devices..... 4
- Alarm Configuration..... 8
- Monitoring VideoXpert Situations..... 11
- Custom (External) Situations..... 12
 - The CustomSituations.xml file 12
- Frequently Asked Questions..... 14
- Pelco Troubleshooting Contact Information..... 14

Overview

The VideoXpert Event Monitor Service is capable of being configured to poll for alarms from various devices, taking the alarm data it gathers and performing various actions when an alarm state change occurs.

Installation

PREREQUISITES

Prior to installation, the following should be installed:

- .NET Framework 4.61 or later
- Microsoft Visual C++ 2015 Redistributables (x86)

VERSIONING AND LICENSING INFORMATION

Version 3.x and above of the service requires VideoXpert version 3.1.x and higher in order to communicate to the VideoXpert system.

As of VideoXpert version 3.1.x, licensing is required to operate the service. The part number for the Event Monitor Service is "VX-EVENT-MON".

PROCEDURE

Note: see **Installation > Prerequisites** for a list of requirements prior to installation. If any of the prerequisites are missing, the service will be unable to run and the installation will fail.

To install the VxEventMonitorService, run the provided `setup.msi` file and follow the on-screen prompts.

The service will install as the local system user. Once installed, you need to complete the configuration (described in **VideoXpert Event Monitor Configuration**), and then either restart the system or manually start the service. On subsequent restarts of the system the service will automatically start.

Default XML Files

In the directory `C:\Program Files (x86)\Pelco\VxEventMonitorService\` there will be several default XML files with the text "default" preceding the file name. These files are examples of XML configuration files required by the service and should be renamed without "default" in the filename once the file has been modified for use. Each configuration file is described below.

VideoXpert Event Monitor Configuration

The VideoXpert Event Monitor Service requires the configuration information contained in the `defaultVxEventMonitorServiceSettings.xml` file to be defined and, as with other files beginning with "default", to be renamed `VxEventMonitorServiceSettings.xml` once the file has been altered for use.

The `VxEventMonitorServiceSettings.xml` file contains the following XML elements:

```
< VxEventMonitorServiceSettings >
    <DebugLevel>1</DebugLevel>
    <VxCoreAddress></VxCoreAddress>
    <VxCorePort></VxCorePort>
    <VxUsername></VxUsername>
    <VxPassword></VxPassword>
```

```
</ VxEventMonitorServiceSettings >
```

Where:

- `DebugLevel` is either 0 for off, 1 for normal debug, 2 will output verbose debug messages. Output may be seen by turning off the service and starting the service from the command line or Windows Explorer.
- `VxCoreAddress` is a string representing the IPV4 Ethernet address of the core. Example: 192.168.1.100.
- `VxCorePort` is the port of the Core the Event Monitor service will be communicating with. Default 443.
- `VxUserName` is a base64 encoded string. Example: "admin" is encoded as YWRtaW4=
- `VxPassword` is a base64 encoded string.

Note that `VxUserName` and `VxPassword` will be encrypted once the service starts. The base64 encoded string is read in by the service, encrypted and written back to the file for security. If the username or password need to change, base64 values can be placed in the file and the service will overwrite the new values with encrypted values on service startup.

Camera Configuration

As of VideoXpert version 1.12, camera numbers may be assigned using the VideoXpert Ops Center.

Monitor Configuration

As of VideoXpert version 1.12, monitor numbers may be assigned using the VideoXpert Ops Center.

Supported Alarm Devices

There are currently three alarm devices that are supported by the Event Monitor Service – the Moxa ioLogik E1212 version v2.5 Build16091211, the Sorhea Maxibus GFence, Hub version V2.5.5 04/01/19 and the IPVideo Corporation Halo Version 1.53. Other variations or versions of these devices may work but have not been tested or verified as supported.

Moxa ioLogik E1212

The Event Monitor Service can poll alarms from and set relays on the Moxa ioLogik E1212 through its Modbus Ethernet interface. Setup of the Moxa ioLogik E1212 must include enabling the Modbus interface by checking the checkbox "Enable Modbus/TCP Slave Protocol" on the User-defined Modbus Addressing Digital inputs/Outputs screen of the Moxa web interface. Address values should be left default.

Alarms polling is initiated every 300 milliseconds.

Sorhea Maxibus GFence

The Event Monitor Service can poll alarms from the Sorhea Maxibus GFence through its Modbus holding registers when Maxibus is set as "Server TCP" as shown below.

TIME SETTING

Hub soft version
V2.5.5 04/01/19

Hub Time
13/01/00 20:55:34

Computer time
24/02/20 09:10:05

SET PRODUCTS TIME

MODBUS

Modbus

Server TCP Client TCP

Slave RTU Master RTU

Unit ID Modbus ...
48

SAVE

Several settings have been added to support Maxibus, including `UnitId` and `StartHolding`. A GFence product attached to COM1 of the Maxibus Hub would require the `StartHolding` address to be set to 0x9066, which is the address of the first alarm. Since Maxibus alarms each have 6 holding registers, the next alarm will be at 0x906c. See the Maxibus documentation for address setup information on other configurations. Note that the `RegisterAddress` of each alarm can also be individually set using the `<RegisterAddress>` element. So, in the scenario above, if it was desired to read the data for the second alarm from the address 0x906d, instead of 0x906c, this can be specified as `<RegisterAddress>0x906d</RegisterAddress>`.

The Event Monitor Service defaults to a bitmask of 0x80 which is looking at the B7 Intrusion bit in the First status word to determine whether or not an alarm condition exists, however, the bitmask may be set using the `<BitMask>` element for each alarm individually in order to support other alarm conditions reported through bits other than B7. If bits B7 and B6 must be set simultaneously for an alarm condition, this may be specified by `<BitMask>0xC0</BitMask>` or `<BitMask>192</BitMask>`.

Polling of Holding registers occurs roughly every 300 milliseconds.

IPVideo Corporation Halo Device

Halo devices must be programmed to send events to the `HaloListenIp` and `HaloListenPort`.

Notifications should be set to where the `HaloListenIp` and `HaloListenPort` match the address and port from the Notifications tab of the Halo Web UI.

When setting up XML for alarms in the `AlarmConfiguration.XML` for a Halo device, the Alarm Name must match the string coming from the Halo device's `%EID%`. For example

```
<Alarm>
  <Name>Aggression</Name>
```


Note that values sent from a Halo device must be delimited with the '|' character. Spaces cannot be used as they may appear in values, such as the `%EID%`.


These strings can be copied and pasted directly into the Notifications settings as shown below.

- `set|%IP|%EID|%THR|%VAL%`
- `reset|%IP|%EID|%THR|%VAL%`
- `heartbeat|%IP%`

Note: only TCP is supported.

VMS Settings

Set string On Off 

Reset String On Off 

Address


Port

Protocol TCP UDP

Save

Above, you can use:
%NAME% - device name
%IP% - ip address
%MAC% - mac address
%EID% - event id
%SOURCE% - data source
%THR% - event threshold
%VAL% - sensor value
%DATE% - local time of event
%TIME% - local time of event

Heart Beat

Message 

Interval Seconds

Address

Port

Protocol TCP UDP

Save

Above, you can use:
%NAME% - device name
%IP% - ip address
%MAC% - mac address
%EVENTS% - list of event states
%DATE% - local date
%TIME% - local time

set|10.221.77.40|Room Temp|80|85

The Halo device is capable of sending both the Threshold and Value of the event trigger. Note that these values can be inserted into the situation injected into VideoXpert by defining the Key/Value pairs with empty values in the Alarm Configuration. This would appear in the AlarmConfiguration.xml as follows (See Alarm Configuration section below for more details).

```
<Situation>
```

```

<AlarmState>1</AlarmState>
<Type>external/EventMonitor/HaloAlarm</Type>
<Property>
  <Key>Threshold</Key>
  <Value></Value>
</Property>
<Property>
  <Key>Value</Key>
  <Value></Value>
</Property>

```

Alarm Configuration

The AlarmConfiguration.xml file is used to define alarm devices and what to do when an alarm state change is found on the alarm device. A file named defaultAlarmConfiguration.xml is provided as an example of what the AlarmConfiguration.xml contents should be.

The AlarmConfiguration.xml file contains the following XML elements:

```

<AlarmConfiguration>
  <!-- IP Address to listen on for Halo Events, if left blank the first
  active NIC found will be used -->
  <HaloListenIp></HaloListenIp>
  <!-- Port to list on for Halo Events -->
  <HaloListenPort></HaloListenPort>

  <AlarmDevice>
    <DeviceIP>192.168.1.1</DeviceIP>
    <DeviceType>ioLogik</DeviceType>
    <NumberInputs>8</NumberInputs>
    <NumberOutputs>8</NumberOutputs>
    <StartInputs></StartInputs>
    <StartOutputs></StartOutputs>
    <StartHolding></StartHolding>
    <UnitId></UnitId>
    <OffLineAlarm>...same elements as Alarm...</OffLineAlarm>

  <Alarm>
    <Name>Alarm 1</Name>
    <Number></Number>
    <RegisterAddress></RegisterAddress>
    <BitMask></BitMask>
    <SourceDeviceId>USE_INTEGRATION_ID</SourceDeviceId>
    <Situation>
      <AlarmState>1</AlarmState>
      <Type>external/EventMonitor/Alarm_active</Type>
      <ValidTimeFilter>
        <DaysOfWeek>Monday,Tuesday</DaysOfWeek>
        <DaysOfMonth>1,15</DaysOfMonth>
        <StartTime>8:30</StartTime>
        <EndTime>17:00</EndTime>
      </ValidTimeFilter>
      <Property>
        <Key>alarm_name</Key>
        <Value>Alarm 1</Value>
      </Property>
      <Command>
        <Name>SaveMonitorState</Name>
        <Monitor>1</Monitor>
      </Command>
      <Command>
        <Name>SetLayout</Name>
        <Monitor>1</Monitor>
        <Layout>2x2</Layout>
      </Command>
    </Situation>
  </Alarm>

```

```

        <Name>DisplayCamera</Name>
        <Monitor>1</Monitor>
        <Cell>1</Cell>
        <Camera>1</Camera>
        <PreviousSeconds>0</PreviousSeconds>
    </Command>
    <Command>
        <Name>GoToPreset</Name>
        <Camera>1</Camera>
        <Preset>PRESET1</Preset>
    </Command>
    <Command>
        <Name>SetRelayState</Name>
        <Relay>1</Relay>
        <State>1</State>
        <DeviceIP>192.168.1.1</DeviceIP>
    </Command>
    <Command>
        <Name>BookMark</Name>
        <Camera>1</Camera>
        <Description>Script 1 Bookmark</Description>
    </Command>
</Situation>
<Situation>
    <AlarmState>0</AlarmState>
    <Type>external/EventMonitor/Alarm_inactive</Type>
    <Property>
        <Key>alarm_name</Key>
        <Value>Alarm 1</Value>
    </Property>
    <Command>
        <Name>RestoreMonitorState</Name>
        <Monitor>1</Monitor>
    </Command>
    <Command>
        <Name>DisconnectCamera</Name>
        <Monitor>1</Monitor>
        <Cell>1</Cell>
    </Command>
</Situation>
</Alarm>
</AlarmDevice>
</AlarmConfiguration>

```

Where:

- `HaloListenIp` sets the IP Address of the integration on which to receive Halo events. If left blank the integration will use the first IP Address found on the integration machine.
- `HostListenPort` set the Port on which to receive Halo events.
- `AlarmDevice` defines a device to poll. This element may be repeated as many times as desired.
- `DeviceIP` is the address of the AlarmDevice.
- `DeviceType` is the type of the AlarmDevice. Currently this must be set to either `ioLogik`, `Maxibus` or `Halo`. Defaults to `ioLogik`.
- `OfflineAlarm` is a special alarm definition for what to do when the Alarm device itself is not communicating. This element is not required and may be removed if unused. When the device is offline the AlarmState is 1, when online the AlarmState is 0. The `OfflineAlarm` is otherwise identical to the `Alarm` element and contains all of the same elements.
- `Alarm` defines an alarm on the AlarmDevice. This element must be repeated for each alarm. If an alarm device has more alarms than Alarm elements, the undefined alarms will be ignored. Note that Alarms are defined in order, so the first Alarm on the Alarm Device should appear first in the XML. If it is desired to skip an Alarm, an empty `<Alarm/>` element should be placed in the XML in place of the alarm to skip.
- `NumberInputs` is the number of alarms the device is able to support or is configured to support. Defaults to 8.
- `NumberOutputs` is the number of relay outputs the device is able to support or is configured to support. Defaults to 8.
- `StartInputs` is the starting address of digital inputs. Defaults to 1.
- `StartOutputs` is the starting address of digital outputs. Defaults to 17.
- `StartHolding` is the starting address of holding registers. Defaults to 36966.

- `UnitId` is the Modbus unit identifier of the device. Defaults to 48.
- `Name` is the friendly name of the alarm.
- `Number` is the number to associate with the alarm.
- `RegisterAddress` specifies a Modbus address at which the alarm resides. Alarms are normally linear in the address space starting from either the `StartInputs` or `StartHolding` address. Currently this value is used only by the Maxibus driver to specify an exact address for the alarm in the Holding registers.
- `Bitmask` is a bitmask to be applied to the values read from the Holding registers to indicate alarm. All bits in the bitmask must be 1 for an alarm to be considered Active. Currently this value is used only by the Maxibus driver.
- `SourceDeviceId` is the Id of the device associated to the alarm. For a pre-defined Situation this would likely be the `deviceId` of a camera that you would like to associate with the alarm when injecting an event into VideoXpert. For an external situation type, this should be "USE_INTEGRATION_ID" to have the service automatically use the `integrationId` that has been assigned by core to fill in the `SourceDeviceId` when an event is injected into VideoXpert.
- `Situation` defines a situation to inject into VideoXpert and a set of commands for a given alarm state. For each Alarm there should be two situations, one for what to do when the `AlarmState` is 1 and one for what to do when the `AlarmState` is 0.
- `AlarmState` is either 1 or 0 for ON/OFF respectively.
- `Type` is the Situation Type. See the VideoXpert documentation for pre-defined types and parameters. External Situations must be defined in the `CustomSituations.xml` file which is described later in this document.
- `ValidTimeFilter` is the time at which the alarm is valid. This element may be removed to indicate the alarm is always valid. If any of the values under `ValidTimeFilter` are defined, the time of the alarm event will be checked against the current local time and only those alarms occurring within the `ValidTimeFilter` will trigger.
- `DaysOfWeek` may be defined and should be one or more of the following separated by commas: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and/or Saturday.
- `DaysOfMonth` may be defined and should consist of integer values ranging from 1 to 31 delimited by commas. Note that while it is allowed to define both `DaysOfWeek` and `DaysOfMonth` simulataneously, this combination of filter will likely result in unexpected results as both will be applied if defined.
- `StartTime` and `EndTime` define daily start and end times for the match to occur. These values consist of an hour 0 – 23 and minute 0 – 59 separated by a colon. For example, 13:30 would indicate 1:30 PM.
- A list of properties may be included, which are key/value pairs. Property keys for pre-defined Situation types are pre-defined and must not be changed. External situations allow for adding any user-defined properties that are desired.
- A list of `Commands` may be included, which are actions that will take place when the alarm changes to the defined `AlarmState`. A `Command` must contain the `Name` element and may contain the elements `Monitor`, `Camera`, `Cell`, `Layout`, `PreviousSeconds`, `Preset`, `Pattern`, `Relay`, `State`, `DeviceIP` or `Description` depending upon the `Command`.
- `Name` is the command name and may contain one of the following values: `SetLayout`, `DisplayCamera`, `DisconnectCamera`, `GoToPreset`, `RunPattern`, `SetRelayState`, `SaveMonitorState`, `RestoreMonitorState` or `BookMark`.
- `Monitor` is the number of the Monitor that the command should target.
- `Layout` is the layout value to use in the command. Valid values are: "1x1", "1x2", "2x1", "2x2", "2x3", "3x2", "3x3", "4x3", "4x4", "5x5", "1+12", "2+8", "3+4", "1+5", "1+7", "12+1", "8+2", "1+1+4", "1+4 (tall)", "1+4 (wide)"
- `Cell` is the number of the cell of the Monitor that the command should target. The cell number is 1 to X where X is the number of cells in the layout. For example, in the 4x4 layout, the cell may be 1 to 16.
- `Camera` is the number of the Camera to be used in the command.
- `PreviousSeconds` is a number of seconds before current time to seek the camera to. If 0 or the element is not present, the camera will be displayed live. To display the camera 5 minutes in the past, set `PreviousSeconds` to 300.
- `Preset` is the preset value to use in the command. An example of a commonly used preset value is "PRESET1".
- `Pattern` is the pattern value to use in the command. An example of a commonly used preset value is "PATTERN1".
- `Relay` is the relay number to use in the command.
- `State` is the state to use in the command. Valid values are 1 or 0.
- `DeviceIP` is the IP address of the device.

- `Description` is a description for a `BookMark`. If this field is empty, the description of the `BookMark` will be "EventMonitor situation: ", followed by the situation type being injected (if any).

In the Alarm configuration above, when Alarm 1 transitions to an `AlarmState` of 1, the external/EventMonitor/Alarm_active situation will be injected into VideoXpert. Then the layout on Monitor 1 will be changed to 2x2, Camera 1 will be displayed in Cell 1 of Monitor 1 and will be moved to Preset 1. The relay state on the alarm device at 192.168.1.1 will also be set to 1.

Note that all the commands are shown in the xml above to outline the required elements of each command except for `RunPattern`. This command requires the following elements:

```
<Command>
  <Name>RunPattern</Name>
  <Monitor>1</Monitor>
  <Camera>1</Camera>
  <Pattern>PATTERN1</Pattern>
</Command>
```

Saving and Restoring Monitor State

Monitor state may be saved and restored using the Commands `SaveMonitorState` and `RestoreMonitorState` defined above. When a monitor state is saved, it is saved until restored. No other saves for that monitor will be accepted until the monitor is restored. This means that if multiple alarms occur which have the `SaveMonitorState` command associated with the same monitor, only the first alarm will save state. Once the monitor's state has been restored, the software will again allow the state to be saved for restoration.

Monitoring VideoXpert Situations

The `EventMonitorService` is capable of being configured to listen for specific VideoXpert events and reacting to the occurrence of these events by executing associated commands. Situations to listen for are configured in the file `C:\Program Files (x86)\Pelco\VxEventMonitorService\Subscriptions.xml`. An example of the XML configuration can be found in the `defaultSubscriptions.xml` file in the same directory.

The `Subscriptions.xml` file contains the following XML elements:

```
<Subscriptions>
  <Situation>
    <AckState></AckState>
    <Type>system/stream_loss </Type>
    <SourceDeviceId></SourceDeviceId>
    <ValidTimeFilter>
      <DaysOfWeek>Monday,Tuesday</DaysOfWeek>
      <DaysOfMonth>1,15</DaysOfMonth>
      <StartTime>8:30</StartTime>
      <EndTime>17:30</EndTime>
    </ValidTimeFilter>
    <Property>
      <Key>data_source_id</Key>
      <Value>uuid:69d5a953-81da-4a04-90c8-6cfcec7f6c65</Value>
    </Property>
    <Command>
      <Name>BookMark</Name>
      <Camera>1</Camera>
      <Description>Device Offline BookMark</Description>
    </Command>
  </Situation>
</Subscriptions>
```

Where:

- `Situation` defines a VideoXpert situation to listen for. All events of the corresponding `Type` will be subscribed to by the service. You may define more than one `Situation` to listen for. You may also add an entry with the same

Situation Type but different properties if you wish to execute different commands based on the properties. For example, if you wish to execute a particular command if a particular device fails.

- `AckState` defines the Acknowledgement state of the event. Leave this blank or undefined if you wish to match on the default setting (“AckNeeded” or “NoAckNeeded”). If you wish to match a particular AckState of the event, you may have one of the following values in `AckState`: “AckNeeded”, “Acked”, “AutoAcked”, “NoAckNeeded”, or “Silenced”.
- `SourceDeviceId` provides a means to match events coming from a particular source device. Not to be confused with the `data_source_id` property which may or may not be the same value for some situations, `SourceDeviceId` is present in all events where the `data_source_id` property is not. This field is useful for situations where the `data_source_id` property is not present such as the “system/device_offline” situation.
- A `ValidTimeFilter` may be imposed to define a valid time for the match to occur. Left blank or removed, indicates that any time the situation occurs it will be processed. If any of the values under `ValidTimeFilter` are defined, the time of the event will be checked against the current local time and only those events matching the values in `ValidTimeFilter` will trigger.
- `DaysOfWeek` may be defined and should be one or more of the following separated by commas: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and/or Saturday.
- `DaysOfMonth` may be defined and should consist of integer values ranging from 1 to 31 delimited by commas. Note that while it is allowed to define both `DaysOfWeek` and `DaysOfMonth` simultaneously, this combination of filter will likely result in unexpected results as both will be applied if defined.
- `StartTime` and `EndTime` define daily start and end times for the match to occur. These values consist of an hour 0 – 23 and minute 0 – 59 separated by a colon. For example, 13:30 would indicate 1:30 PM.
- The `Property` element may be repeated for each property you desire to match. Each Situation event from VideoXpert will have a set of properties that can be matched. The string in the Key element must match exactly where the string in the Value element is considered matching if contained within the Situation received from VideoXpert. In the example above, this would allow you to enter a partial uuid for the device that experience stream loss in order for a match to occur. Care should be taken to ensure that enough of the uuid is entered to distinguish the device from other devices on the system.
- The `Command` element defines actions to take when the Situation occurs. The list of acceptable commands is identical to that defined in the AlarmConfiguration section. The `Command` element may be repeated as many times as desired.

Note that for each event Situation received from VideoXpert, you may define multiple Situations that match. Each matching Situation will have the corresponding Commands executed.

Custom (External) Situations

It is possible to define custom situations, referred to as external situations, and place the created external situation type into the Alarm `<situationType>` of an entry in the `AlarmConfiguration.xml`.

There are several ways to view External Events in VideoXpert: Using VideoXpert Toolbox navigate to the “Events” tab or you can use the Event Viewer plugin within Ops Center.

Note that it is possible to modify External Situation properties within Toolbox, however, these settings will be overwritten by settings within the `CustomSituations.xml` file (below) each time the service restarts.

THE CUSTOMSITUATIONS.XML FILE

If the event is a custom event to VideoXpert, as configured above, the custom situation must be defined so that it can be sent through to the VideoXpert system. Custom Situations are defined in the file `C:\Program Files`

`(x86)\Pelco\VxEventManagerService\CustomSituations.xml`.

The `CustomSituations.xml` file contains the following XML elements:

```
<CustomSituations>
  <CustomSituation>
```

```

<SituationType>external/EventMonitor/Alarm_active</SituationType>
<SourceDeviceId></SourceDeviceId>
<Name>Alarm Active</Name>
<Severity>1</Severity>
<Log>1</Log>
<Notify>1</Notify>
<DisplayBanner>1</DisplayBanner>
<ExpandBanner>0</ExpandBanner>
<Audible>1</Audible>
<AckNeeded>0</AckNeeded>
<AutoAcknowledge>10</AutoAcknowledge>
<SnoozeIntervals>60,300,600</SnoozeIntervals>
</CustomSituation>
<CustomSituation>
  <SituationType>external/EventMonitor/Alarm_inactive</SituationType>
  <SourceDeviceId></SourceDeviceId>
  <Name>Alarm InActive</Name>
  <Severity>1</Severity>
  <Log>1</Log>
  <Notify>1</Notify>
  <DisplayBanner>1</DisplayBanner>
  <ExpandBanner>0</ExpandBanner>
  <Audible>1</Audible>
  <AckNeeded>0</AckNeeded>
  <AutoAcknowledge>10</AutoAcknowledge>
</CustomSituation>
</CustomSituations>

```

Where:

- `SituationType` is a string defining the situation type. Must start with “external/” and should follow the rules defined in the VideoXpert protocol documentation.
- `SourceDeviceId` may be left blank or may be defined in order to act as an optional constraint on the source of events for this Situation. If specified, any events matching the Situation type MUST also match this `SourceDeviceId` in order for the Situation to apply. “USE_INTEGRATION_ID” may be used to have the service automatically use the `integrationId` that has been assigned by core to fill in the `SourceDeviceId`
- `Name` is a descriptive string of the situation.
- `Severity` is an integer from 1 – 10.
- `Log` is a Boolean (0 or 1), telling VideoXpert to Log the situation or not.
- `Notify` is a Boolean, telling VideoXpert to Notify users of the situation or not.
- `DisplayBanner` is a Boolean, telling VideoXpert to display a banner in Ops Center or not.
- `ExpandBanner` is a Boolean, telling VideoXpert to expand the banner in Ops Center or not.
- `Audible` is a Boolean, telling VideoXpert to have an audible indication of the event or not.
- `AckNeeded` is a Boolean, telling VideoXpert that an Ack is needed or not.

- `AutoAcknowledge` is an integer number of seconds to automatically acknowledge the alarm.
- `SnoozeIntervals` is an optional parameter. The default is (60, 300, 600). If you do not wish to change the `SnoozeIntervals`, leave it blank or remove the element completely.

Frequently Asked Questions

- **How do I view the logs?**

You can get debug information by the doing the following: Set the DebugLevel > 0 in the configuration file and run dbgView.exe. This should be able to capture and log any output from the service. We simple write to standard debug and there are utilities that can capture and log standard debug.

- **How can I view the alarms in the VideoXpert Event Viewer?**

Typically, events injected from this integration will have a type of "External". A user can input any type of situation they desire, though it should be "External" since it is coming from an integration. Users can filter the events by selecting the "External" event category in the filters area.

- **Can I install the Event Monitor Service on the VideoXpert Core server?**

Installing and running on VideoXpert Core is not recommended. Users should use discretion to not impact the performance of their core system components. It is recommended that the Event Monitor Service be installed on another server. Please refer to the installation section of this document for more information.

- **Can I trigger a relay on devices other than the relays on the External I/O device?**

In this version of the Event Monitor Service it is not possible to trigger relays on other devices within the VideoXpert system other than those on the external I/O device.

Pelco Troubleshooting Contact Information

If the instructions provided fail to solve your problem, contact Pelco Product Support at 1-800-289-9100 (USA and Canada) or +1-559-292-1981 (international) for assistance.

Year	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											
38											
39											
40											
41											
42											
43											
44											
45											
46											
47											
48											
49											
50											
51											
52											
53											
54											
55											
56											
57											
58											
59											
60											
61											
62											
63											
64											
65											
66											
67											
68											
69											
70											
71											
72											
73											
74											
75											
76											
77											
78											
79											
80											
81											
82											
83											
84											
85											
86											
87											
88											
89											
90											
91											
92											
93											
94											
95											
96											
97											
98											
99											
100											



USA & Canada Tel (800) 289-9100 Fax (800) 289-9150
International Tel +1 (559) 292-1981 Fax +1 (559) 348-1120
www.pelco.com www.pelco.com/community