



by **Schneider** Electric

VideoXpert Event Monitor Service

Contents

- Overview3
- Installation3
 - Prerequisites3
 - Procedure3
- Default XML Files3
- VideoXpert Event Monitor Configuration3
- Camera Configuration4
- Monitor Configuration4
- Alarm Configuration.....4
- Monitoring VideoXpert Situations6
- Custom (External) Situations.....7
 - The CustomSituations.xml file.....7
- Supported Alarm Devices.....9
- Frequently Asked Questions9
- Pelco Troubleshooting Contact Information9

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)

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>
    <VideoXpertCoreAddress></VideoXpertCoreAddress>
    <VideoXpertCorePort></VideoXpertCorePort>
    <VideoXpertUsername></VideoXpertUsername>
    <VideoXpertPassword></VideoXpertPassword>
</ 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.
- `VideoXpertCoreAddress` is a string representing the IPV4 Ethernet address of the core. Example: 192.168.1.100.
- `VideoXpertCorePort` is the port of the Core the Event Monitor service will be communicating with. Default 443.
- `VideoXpertUserName` is a base64 encoded string. Example: "admin" is encoded as YWRtaW4=
- `VideoXpertPassword` is a base64 encoded string.

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.

Alarm Configuration

The `AlarmConfiguration.xml` file is used to define alarm devices that will be polled by the service and what to do when an alarm state change is found. 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>
  <AlarmDevice>
    <DeviceIP>192.168.1.1</DeviceIP>
    <DeviceType>ioLogik E1212</DeviceType>
    <NumberInputs>8</NumberInputs>
    <NumberOutputs>8</NumberOutputs>
    <Alarm>
      <Name>Alarm 1</Name>
      <SourceDeviceId>USE_INTEGRATION_ID</SourceDeviceId>
      <Situation>
        <AlarmState>1</AlarmState>
        <Type>external/EventMonitor/Alarm_active</Type>
        <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>
        <Command>
          <Name>DisplayCamera</Name>
          <Monitor>1</Monitor>
          <Cell>1</Cell>
          <Camera>1</Camera>
          <PreviousSeconds>0</PreviousSeconds>
        </Command>
        <Command>
          <Name>GoToPreset</Name>
          <Monitor>1</Monitor>
          <Camera>1</Camera>
          <Preset>PRESET1</Preset>
        </Command>
      </Situation>
    </Alarm>
  </AlarmDevice>
</AlarmConfiguration>
```

```

        <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:

- `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 `ioLogik E1212`.
- `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.
- `Name` is the friendly name of the alarm.
- `Number` is the number to associate with the alarm.
- `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. 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.
- 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 contains 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/device_offline</Type>
    <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>
```

```
</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”.
- 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 went offline 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.

Supported Alarm Devices

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.

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.

REVISION HISTORY

Manual Title	Date	Comments
Event Monitor Service	05/17	v2.0.2 updates.
Event Monitor Service	02/17	Original version.

Pelco, the Pelco logo, and other trademarks associated with Pelco products referred to in this publication are trademarks of Pelco, Inc. or its affiliates. ONVIF and the ONVIF logo are trademarks of ONVIF Inc. All other product names and services are the property of their respective companies. Product specifications and availability are subject to change without notice.

© Copyright 2017, Pelco, Inc.
All rights reserved.



Pelco by Schneider Electric 3500 Pelco Way Clovis, California 93612-5699 United States
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