PI System Status Monitoring

AVEVA Group plc
High Cross, Madingley Road
Cambridge CB3 0HB, UK
Tel +44 (0)1223 556655
Fax +44 (0)1223 556666

aveva.com

Table of contents

1. Learning Objectives

The PI System can be used to monitor other PI System implementations with analytics, event frames, and notifications. This course will walk you through using the PI System Monitoring example kit to make a digital twin of PI System assets. Learn where to find PI System diagnostic data streams and how to do simple analytics to alert you of problems. Learn how to provide immediate instruction relative to the symptoms using email messages.

This course is designed for PI System Administrators and PI support people who have basic technical knowledge of the main PI System components (Data Archive, Asset Framework, Asset Analytics (including Event Frames) and Notifications) and are familiar with the PI Interface for Performance Monitor and one or more common real time data interfaces (PI OPC, PI RDBMS, PI to PI, etc.). Students should already know the basics of how the PI System components and interfaces function technically and where interdependencies exist and be familiar with how to deploy AF templates and configure an AF hierarchy in PI System Explorer. Students should also understand Asset Analytics and how Event Frames and Notifications work. It is highly recommended students have previously taken the PI System Administration for IT Professionals course.

2. PI System Software Components

The VM (virtual machines) used for this course have the following PI System software components installed:

Software	Version
Data Archive 2018 SP3 Patch 2	3.4.440.477
PI AF Server 2018 SP3 Patch 3	2.10.9.593
PI Analysis Service 2018 SP3	2.10.6.195
PI Notifications Service 2018 SP3 Patch 3	2.10.9.593
PI Buffer Subsystem 2018 SP2 Patch 2	4.9.0.37
PI Interface for OPC DA Read-Only	2.7.1.41
PI Interface for Performance Monitor	2.2.0.38
PI Interface for Ping	2.1.2.49
PI Interface for TCP Response	1.3.0.47
PI Integrator for Business Analytics 2020 R2	2.4.600.725
PI Connector for OPC UA	1.3.1.142

PI Connector for UFL	1.3.2.139
PI Vision 2021	3.6.0.0
Microsoft SQL Server	2019 (PIAF01) & 2017 (PIMONITOR01)
Microsoft Excel 2016 (including PI DataLink and PI Builder plug-ins)	16.0.4266.1001

3. Introduction

Time series sensor data is an extremely valuable asset. It is the final result of your instrumentation investment. Poor data quality directly impacts your bottom line. Every PI System is unique in the wide variety of platforms and applications it touches. Data streams can be impacted at any time due to application, platform, and network changes. To be a leader in capturing good data and performing high quality analytics with your PI System you need the help of diagnostic monitoring.

Scenario

You work in IT/Application Support at NuGreen and have been assigned as the PI System Administrator for a recently installed PI System. You have just returned from the PI System Management course and have a basic technical understanding of the main components in your PI System environment. The business has asked you to implement some type of proactive monitoring of the PI System deployment so the support teams have a better view into the current technical status of the PI System components and can be aware of issues and address problems earlier, resulting in less impact to end users.

Since you use the PI System to monitor all your plant equipment, you figure you might as well use the PI System to monitor your PI System equipment similarly! You contact your Customer Success Manager who provides you with the PI System Monitoring (PSM) example kit – a set of example AF templates that can be used as a starting point for building a digital twin of your PI System deployment. You follow this course to learn to implement the PSM example kit.

The course environment has been configured to represent NuGreen's PI System deployment. This PI System contains 4 servers and includes:

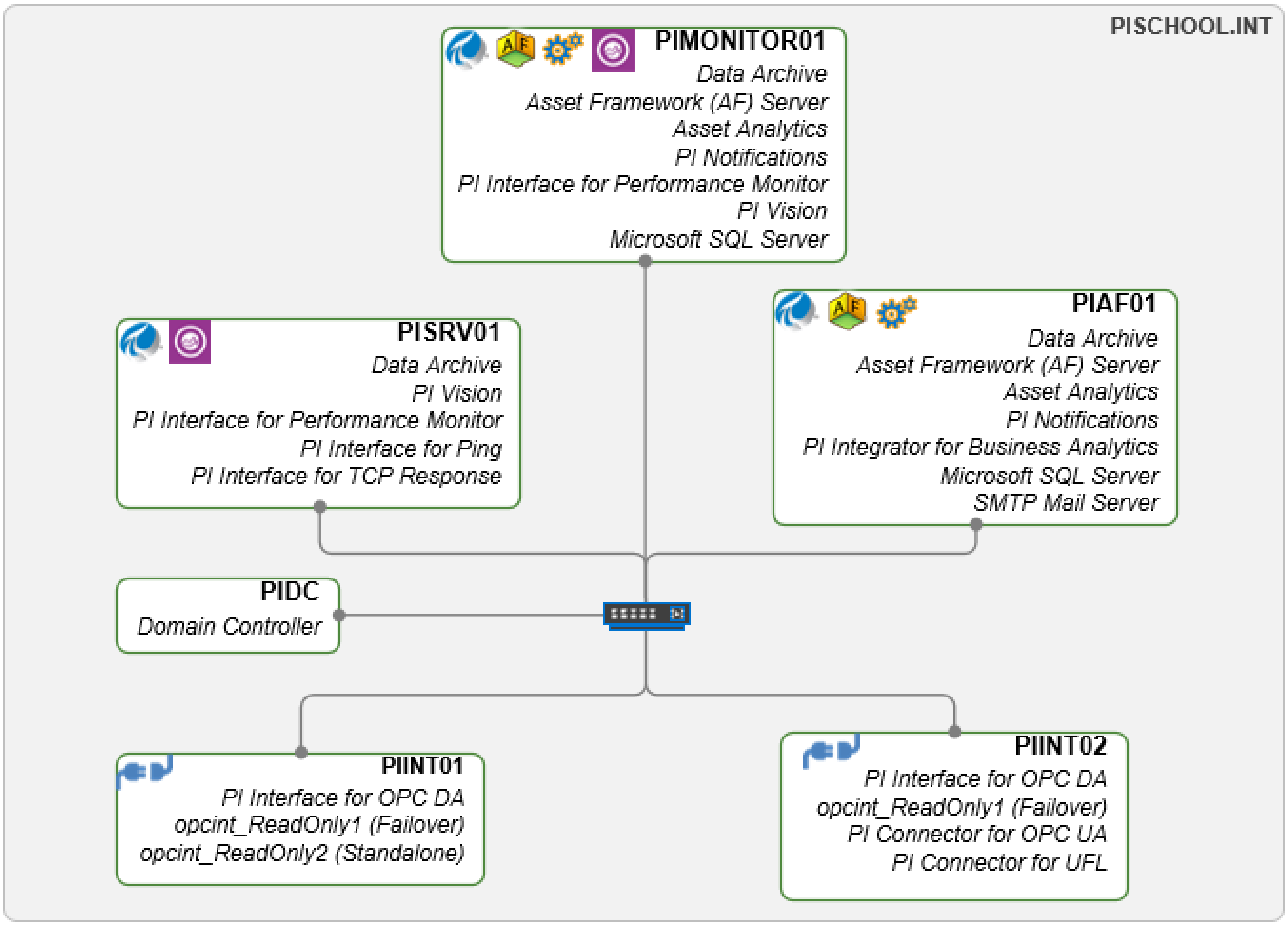
- Data Archive collective on PISRV01 (primary) and PIAF01 (secondary)
- AF Server, Asset Analytics, PI Notifications, and SQL Server on PIAF01
- PI Integrator for Business Analytics on PIAF01
- PI Interfaces:
    - Ping and TCP Response on PISRV01
    - OPC DA (standalone and failover) on PIINT01
    - OPC DA (failover) on PIINT02
- PI Connectors:
    - OPC UA and UFL on PIINT02

A second PI System has been configured on a single server (PIMONITOR01) as the monitor and includes:

- Data Archive (standalone)
- AF Server, Asset Analytics, PI Notifications
- PI Interface for Performance Monitor
- PI Vision
- SQL Server

You can be provided links to access each server via a web browser remote desktop connection. Most of your work during the course will be done on the monitoring PI System.

Architecture diagram



Your PI landscape has the following attributes:

- All PI System components in your landscape (listed above) have been installed and configured already, and most should be actively running. No system monitoring PI Tags have been created yet (apart from the answer database).
    - The PI Interface for Performance Monitor on PIMONITOR01 is already configured and is using point source: **PIMONITOR01_PERF01**. (NOTE: there is a second instance of PI Interface for Performance Monitor used by the answer database – it uses point source PIMONITOR01_PERF02. You will ignore this instance in your course.)

- There is an answer AF database (**PSM Course Answer**) in your system, including an AF hierarchy, system monitoring PI Tags, and an email account for notifications. This answer database monitors the same components you will create monitoring for in the course (and more). We will refer to this for certain examples and explanations, but you will be creating items in your own AF database.
    - The database you will use has already been created, and is called **PSM Course Student**. You will import the PSM example kit, and use it to create an element hierarchy and system monitoring PI Tags. Your database uses a separate email account for notifications.
- There is an existing SMTP mail server (piaf01.pischool.int) which has already been configured. You can access email via Outlook on PIMONITOR01. The email address for your account on all course systems is: ***student01@pischool.int*****.** (NOTE: there is a second email account used by the answer database: *student02@pischool.int* – we will only use this for certain examples). The password for both accounts in Outlook is the same as the password to log onto your machines.
- There is a Domain Controller in your landscape which is already configured – you will ignore this for the course. Windows security is used.

Business objective

A set of example AF templates for PI System Monitoring will be provided, and students will learn to configure an associated AF hierarchy using PI System Explorer. PI Interface for Performance Monitor will be used to capture data from Windows performance counters into PI Tags for monitoring. Students will simulate PI System issues and explore the monitoring capabilities of PI System Explorer and PI Notifications. Each exercise can be completed with the tools provided in class.

The content in this course can be provided as a guided and customized workshop, to take you through deploying the example kit in your own environment. If you are interested in a PSM workshop, please contact your Customer Success Manager.

Note to students

This course only covers a few failure scenarios, while the PSM example kit templates provide monitoring for many different issues. Not all templates are covered in the exercises, and for templates that are covered not all attributes, analyses, and notifications are discussed. Students are encouraged to review these on their own to determine how they work and what other issues are monitored. See the following sections for optional exercises to cover a few extra failure scenarios, and high-level overviews of templates not used in the exercises:

- *Optional exercise 1: Server – high processor utilization*
- *Optional exercise 2: PI Buffer subsystem – low buffer queue capacity*
- *Other Answer Components*

4. Exercise 1: Environment assessment & database configuration

Exercise objectives

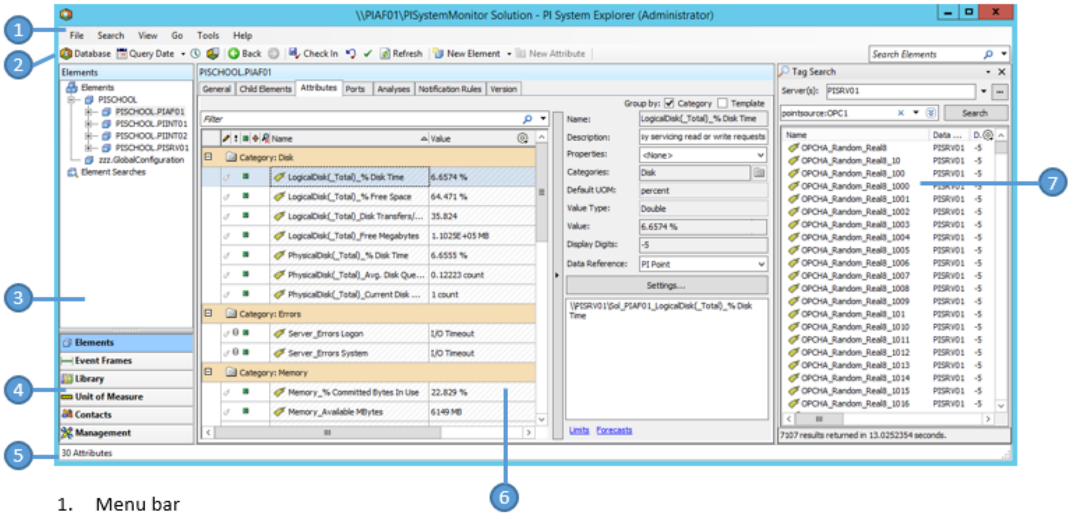
- Understand your PI System environment
- Explore the PI System Monitoring example kit
- Import the example kit and create configuration components

Scenario

You will begin by exploring your PI landscape. You will review the library to understand the components in it. You will import the example kit into your own database and create the required configuration components.

Approach

1. Review the PI architecture diagram provided in the Scenario at the beginning of this course manual. This shows all the servers in your PI landscape and which components are installed on each.
   a. How many domains are used?
   b. How many servers are there? What are their machine names?
   c. Which PI System components are installed and on which machines?
2. Log onto PIMONITOR01 using a web browser.
3. Open PI System Explorer (icon on Taskbar) – it will automatically open the **PSM Course Answer** database.
   a. Before you proceed you should be familiar with the various areas of PI System Explorer:



30 Attributes

Name	Value
Category: Disk	
LogicalDisk(_Total)_% Disk Time	6.6574 %
LogicalDisk(_Total)_% Free Space	64.471 %
LogicalDisk(_Total)_Disk Transfers/...	35.824
LogicalDisk(_Total)_Free Megabytes	1.1025E+05 MB
PhysicalDisk(_Total)_% Disk Time	6.6555 %
PhysicalDisk(_Total)_Avg. Disk Que...	0.12223 count
PhysicalDisk(_Total)_Current Disk ...	1 count
Category: Errors	
Server_Errors Logon	I/O Timeout
Server_Errors System	I/O Timeout
Category: Memory	
Memory_% Committed Bytes In Use	22.829 %
Memory_Available MBytes	6149 MB

Name: LogicalDisk(_Total)_% Disk Time
Description: ıy servicing read or write requests
Properties: <None>
Categories: Disk
Default UOM: percent
Value Type: Double
Value: 6.6574 %
Display Digits: -5
Data Reference: PI Point

\\PISRV01\Sol_PIAF01_LogicalDisk(_Total)_% Disk Time

Server(s): PISRV01
pointsource:OPC1

Name	Data ...	D.
OPCHA_Random_Real8	PISRV01	-5
OPCHA_Random_Real8_10	PISRV01	-5
OPCHA_Random_Real8_100	PISRV01	-5
OPCHA_Random_Real8_1000	PISRV01	-5
OPCHA_Random_Real8_1001	PISRV01	-5
OPCHA_Random_Real8_1002	PISRV01	-5
OPCHA_Random_Real8_1003	PISRV01	-5
OPCHA_Random_Real8_1004	PISRV01	-5
OPCHA_Random_Real8_1005	PISRV01	-5
OPCHA_Random_Real8_1006	PISRV01	-5
OPCHA_Random_Real8_1007	PISRV01	-5
OPCHA_Random_Real8_1008	PISRV01	-5
OPCHA_Random_Real8_1009	PISRV01	-5
OPCHA_Random_Real8_101	PISRV01	-5
OPCHA_Random_Real8_1010	PISRV01	-5
OPCHA_Random_Real8_1011	PISRV01	-5
OPCHA_Random_Real8_1012	PISRV01	-5
OPCHA_Random_Real8_1013	PISRV01	-5
OPCHA_Random_Real8_1014	PISRV01	-5
OPCHA_Random_Real8_1015	PISRV01	-5
OPCHA_Random_Real8_1016	PISRV01	-5

7107 results returned in 13.0252354 seconds.

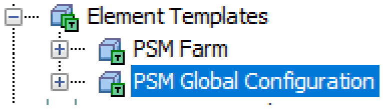
1. Menu bar
2. Toolbar
3. Browser
4. Navigator
5. Status bar
6. Viewer
7. Palette

b. Select **Library** in the navigator (lower left) to open the Library in the browser.

  i. You will see a list of templates from the answer listed under **Element Templates** in the browser

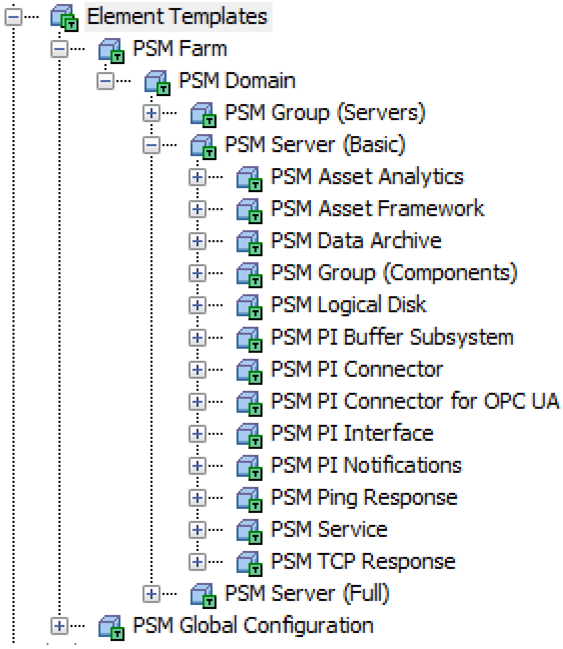
  ii. Right-click on **Element Templates** and select **Arrange By > Arrange By Template References**.

  iii. Note the **PSM Global Configuration** element template. You will see more of this when you set up your database, but for now just know just know that it is a required part of the structure and contains several necessary configuration attributes.



- Element Templates
  - PSM Farm
  - PSM Global Configuration

  iv. Expand the tree under **PSM Farm > PSM Domain > PSM Server (Basic)** and you will see several types of element templates for PI System components under PSM Server (Basic).



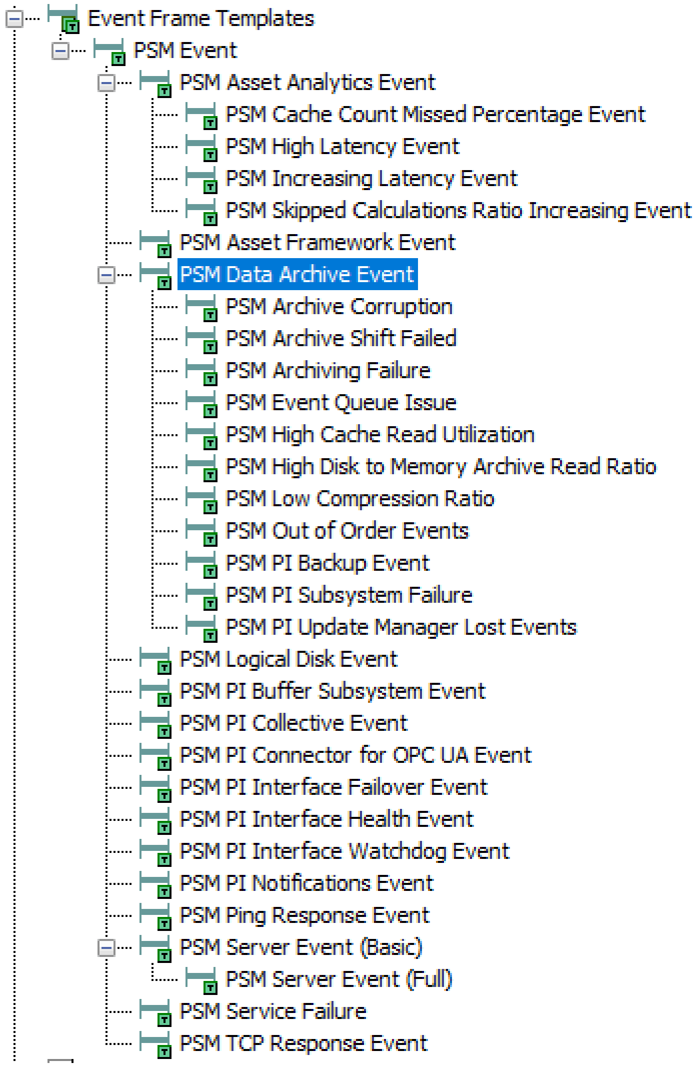
- Element Templates
  - PSM Farm
    - PSM Domain
      - PSM Group (Servers)
      - PSM Server (Basic)
        - PSM Asset Analytics
        - PSM Asset Framework
        - PSM Data Archive
        - PSM Group (Components)
        - PSM Logical Disk
        - PSM PI Buffer Subsystem
        - PSM PI Connector
        - PSM PI Connector for OPC UA
        - PSM PI Interface
        - PSM PI Notifications
        - PSM Ping Response
        - PSM Service
        - PSM TCP Response
      - PSM Server (Full)
  - PSM Global Configuration

You can see this uses a type of ‘container’ system whereby PI System components are part of servers, which are part of domains, which are part of farms.

1. If you expand **PSM Server (Full)** you will see the same list of templates – that is because PSM Server (Full) is derived from PSM Server (Basic) (it adds a few items). This means you can use either of the two PSM Server templates to represent a server.
2. There are also group templates both above and below PSM Server (*) – PSM Group (Servers) above is used to organize a group of servers, while PSM Group (Components) below is used to organize a group of PI System components. These are both just additional organizational templates and are optional.
3. Apart from the PSM Global Configuration element, the only required element in a PSM database is PSM Server (*) as it contains certain attributes used in PI Tag naming which are also inherited by its child elements. The PSM Farm, PSM Domain, and PSM Group (*) templates are intended for organizational purposes but are not required for the template library to work.

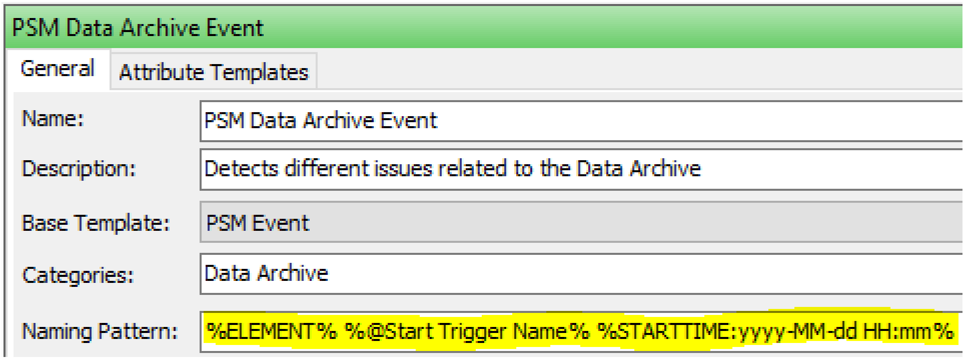
v. Next expand the **Event Frame Templates**. Right-click and select **Arrange By > Arrange by Template Inheritance**.

1. Expand the **PSM Event** and then the **PSM Data Archive Event** to see the various event frames used for monitoring PI System components.



- Event Frame Templates
  - PSM Event
    - PSM Asset Analytics Event
      - PSM Cache Count Missed Percentage Event
      - PSM High Latency Event
      - PSM Increasing Latency Event
      - PSM Skipped Calculations Ratio Increasing Event
    - PSM Asset Framework Event
    - PSM Data Archive Event
      - PSM Archive Corruption
      - PSM Archive Shift Failed
      - PSM Archiving Failure
      - PSM Event Queue Issue
      - PSM High Cache Read Utilization
      - PSM High Disk to Memory Archive Read Ratio
      - PSM Low Compression Ratio
      - PSM Out of Order Events
      - PSM PI Backup Event
      - PSM PI Subsystem Failure
      - PSM PI Update Manager Lost Events
    - PSM Logical Disk Event
    - PSM PI Buffer Subsystem Event
    - PSM PI Collective Event
    - PSM PI Connector for OPC UA Event
    - PSM PI Interface Failover Event
    - PSM PI Interface Health Event
    - PSM PI Interface Watchdog Event
    - PSM PI Notifications Event
    - PSM Ping Response Event
    - PSM Server Event (Basic)
      - PSM Server Event (Full)
    - PSM Service Failure
    - PSM TCP Response Event

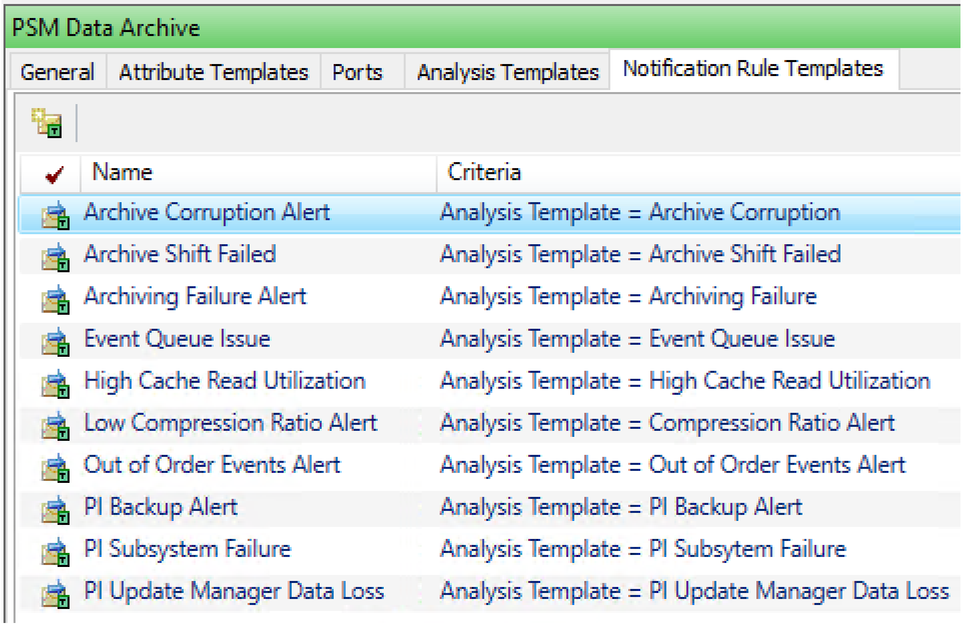
2. Select the **PSM Data Archive Event** template and look at the **General** tab in the viewer. Note that the **Naming Pattern** contains various aspects of the element, start trigger, and time. This is important because it provides key information right in the name of the event frame – which component is having issues, what triggered the issue, and when.



PSM Data Archive Event

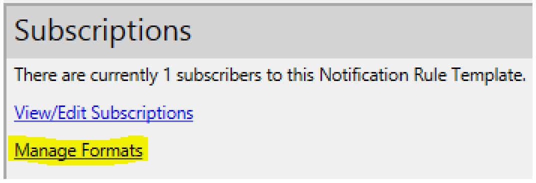
Name:	PSM Data Archive Event
Description:	Detects different issues related to the Data Archive
Base Template:	PSM Event
Categories:	Data Archive
Naming Pattern:	%ELEMENT% %@Start Trigger Name% %STARTTIME:yyyy-MM-dd HH:mm%

vi. Back under the **Element Templates**, select **PSM Data Archive** and select the **Notification Rule Templates** tab in the viewer.



✓	Name	Criteria
	Archive Corruption Alert	Analysis Template = Archive Corruption
	Archive Shift Failed	Analysis Template = Archive Shift Failed
	Archiving Failure Alert	Analysis Template = Archiving Failure
	Event Queue Issue	Analysis Template = Event Queue Issue
	High Cache Read Utilization	Analysis Template = High Cache Read Utilization
	Low Compression Ratio Alert	Analysis Template = Compression Ratio Alert
	Out of Order Events Alert	Analysis Template = Out of Order Events Alert
	PI Backup Alert	Analysis Template = PI Backup Alert
	PI Subsystem Failure	Analysis Template = PI Subsytem Failure
	PI Update Manager Data Loss	Analysis Template = PI Update Manager Data Loss

1. Select the **Archive Corruption Alert** rule and click **Manage Formats** under Subscriptions on the right in the Viewer



Subscriptions

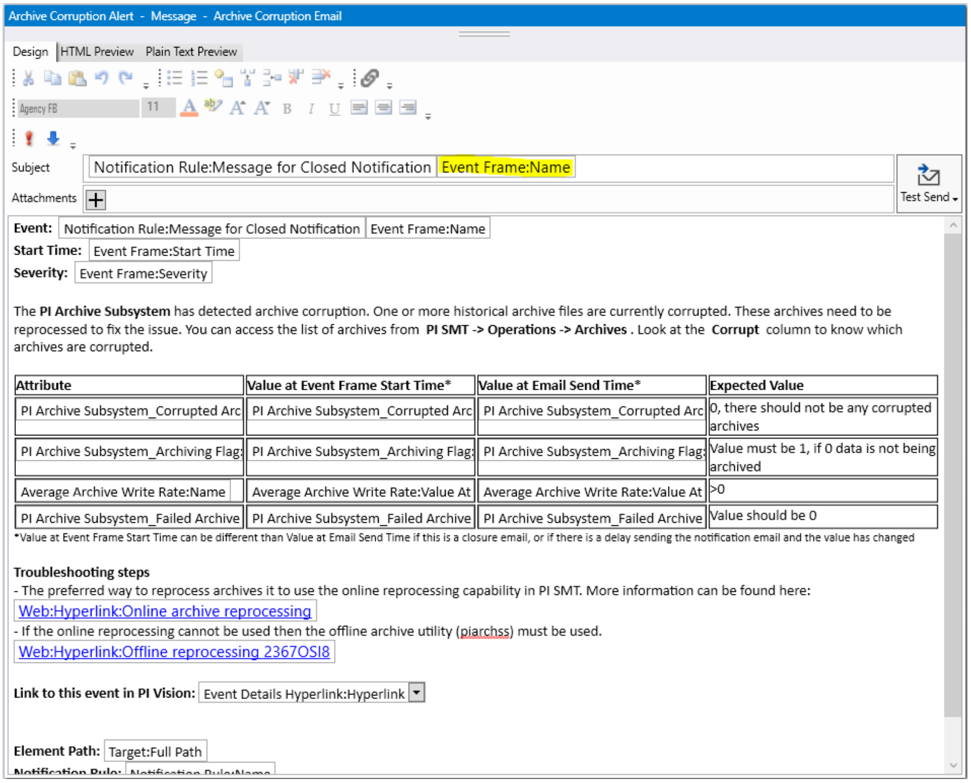
There are currently 1 subscribers to this Notification Rule Template.

View/Edit Subscriptions

Manage Formats

2. In the popup window you will see the design of the email that will be sent out when this notification is triggered by the event frame. Some key things to note are the inclusion of the **Event Frame:Name** in the subject – this narrows focus to the specific issue. As you saw previously, the **Start Trigger Name** is included in

the event frame name to identify the specific cause of the issue.



Archive Corruption Alert - Message - Archive Corruption Email
Design
HTML Preview
Plain Text Preview
Agency FB
11
Test Send

Subject: Notification Rule:Message for Closed Notification Event Frame:Name

Attachments

Event: Notification Rule:Message for Closed Notification Event Frame:Name
Start Time: Event Frame:Start Time
Severity: Event Frame:Severity

The **PI Archive Subsystem** has detected archive corruption. One or more historical archive files are currently corrupted. These archives need to be reprocessed to fix the issue. You can access the list of archives from **PI SMT -> Operations -> Archives** . Look at the **Corrupt** column to know which archives are corrupted.

Attribute	Value at Event Frame Start Time*	Value at Email Send Time*	Expected Value
PI Archive Subsystem_Corrupted Arc	PI Archive Subsystem_Corrupted Arc	PI Archive Subsystem_Corrupted Arc	0, there should not be any corrupted archives
PI Archive Subsystem_Archiving Flag:	PI Archive Subsystem_Archiving Flag:	PI Archive Subsystem_Archiving Flag:	Value must be 1, if 0 data is not being archived
Average Archive Write Rate:Name	Average Archive Write Rate:Value At	Average Archive Write Rate:Value At	>0
PI Archive Subsystem_Failed Archive	PI Archive Subsystem_Failed Archive	PI Archive Subsystem_Failed Archive	Value should be 0

*Value at Event Frame Start Time can be different than Value at Email Send Time if this is a closure email, or if there is a delay sending the notification email and the value has changed

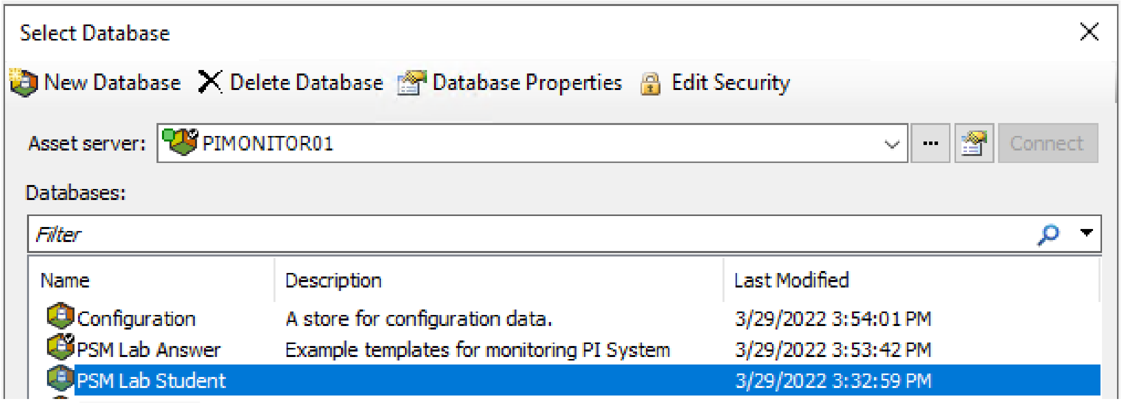
Troubleshooting steps
- The preferred way to reprocess archives it to use the online reprocessing capability in PI SMT. More information can be found here:
Web:Hyperlink:Online archive reprocessing
- If the online reprocessing cannot be used then the offline archive utility (piarchss) must be used.
Web:Hyperlink:Offline reprocessing 2367OSI8

Link to this event in PI Vision: Event Details Hyperlink:Hyperlink

Element Path: Target:Full Path
Notification Rule: Notification Rule:Name

3. Click **OK** to close the email window.

4. Select **Elements** in the navigator to bring up **Elements** in the browser - you will see the element structure in the answer database. You can expand various levels to see what is in there. The answer database has many more elements than what you will create in this course. For further information on these elements see the optional exercises and *Other Answer Components* sections of this course manual and view the templates and elements in the answer database.
5. A new PI System Monitoring database has already been created for you to use. In PI System Explorer, click Database on the top left of the toolbar.
   a. Select the **PSM Course Student** database in the list and click **OK**

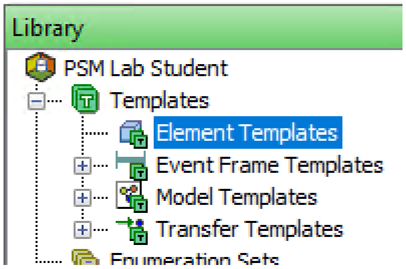


Asset server: PIMONITOR01

Databases:

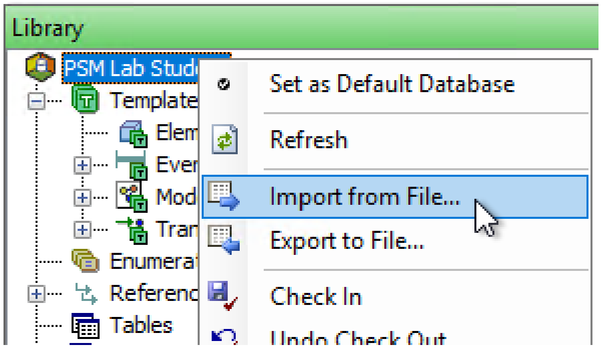
Name	Description	Last Modified
Configuration	A store for configuration data.	3/29/2022 3:54:01 PM
PSM Lab Answer	Example templates for monitoring PI System	3/29/2022 3:53:42 PM
PSM Lab Student		3/29/2022 3:32:59 PM

   b. Select **Library** in the navigator to bring up the library in the browser. If you click the + next to Element Templates to expand you will see there are no templates in the database yet. You will need to import the example kit.

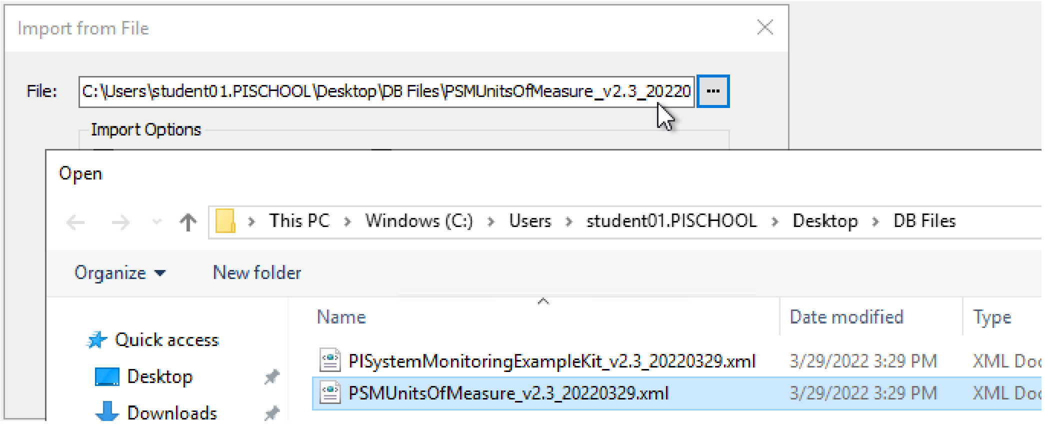


c. Prior to importing new templates into a database you should ensure any new units of measure have been imported, or you will see errors when importing the templates. Keep in mind that the Units of Measure database is global for an entire AF System, and all AF databases have access to the same Units of Measure database. Technically the below step is not necessary for the course because we've already loaded the new units of measure for the answer database, but remember to do this if you decide to deploy the PSM example kit on another PI System.

   i. Ensure the **PSM Course Student** database is selected at the top of the library list in the browser. Right-click on the database name and select **Import from File…**



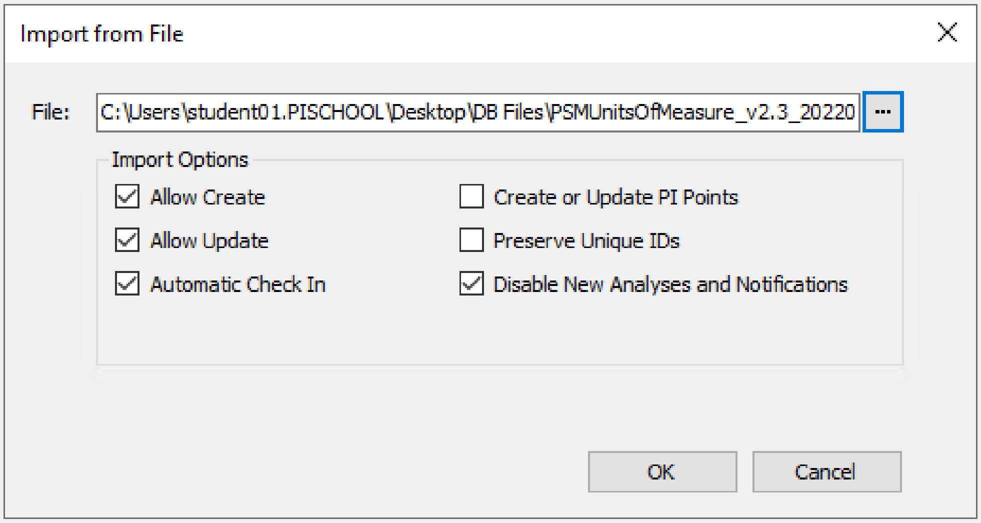
   ii. Click on the ellipses (…) button to the right of the file path, and select the PSMUnitsOfMeasure_vN.N_YYYYMMDD.xml file.



File: C:\Users\student01.PISCHOOL\Desktop\DB Files\PSMUnitsOfMeasure_v2.3_20220

Name	Date modified	Type
PISystemMonitoringExampleKit_v2.3_20220329.xml	3/29/2022 3:29 PM	XML Doc
PSMUnitsOfMeasure_v2.3_20220329.xml	3/29/2022 3:29 PM	XML Doc

iii. The following import options should be selected. Click **OK**.



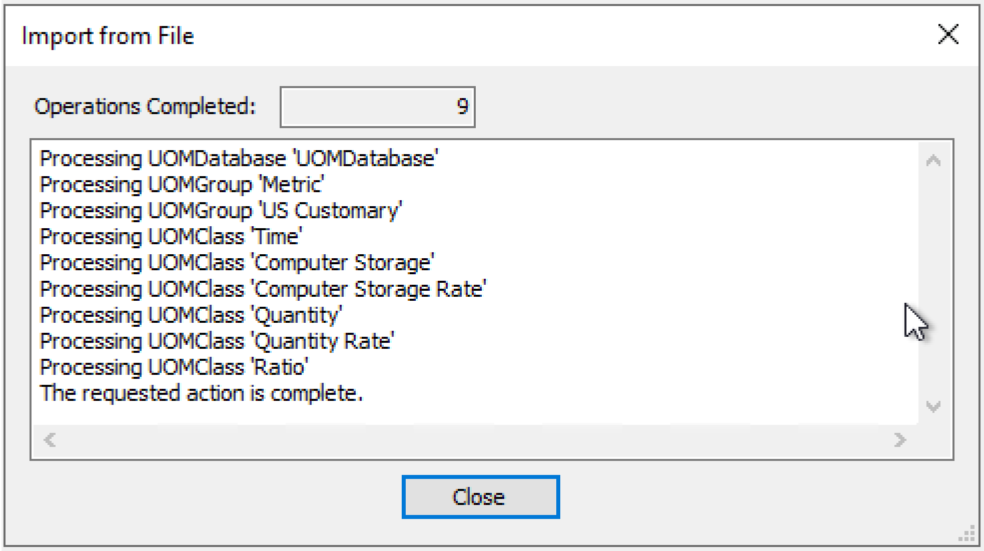
Import from File

File: C:\Users\student01.PISCHOOL\Desktop\DB Files\PSMUnitsOfMeasure_v2.3_20220

Import Options

☑ Allow Create
☑ Allow Update
☑ Automatic Check In
☐ Create or Update PI Points
☐ Preserve Unique IDs
☑ Disable New Analyses and Notifications

iv. After it is done importing you should see a message 'The requested action is complete'. You can close the import window.



Import from File

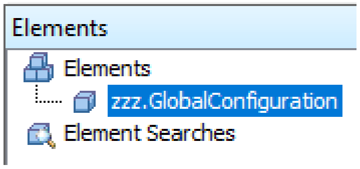
Operations Completed: 9

```
Processing UOMDatabase 'UOMDatabase'
Processing UOMGroup 'Metric'
Processing UOMGroup 'US Customary'
Processing UOMClass 'Time'
Processing UOMClass 'Computer Storage'
Processing UOMClass 'Computer Storage Rate'
Processing UOMClass 'Quantity'
Processing UOMClass 'Quantity Rate'
Processing UOMClass 'Ratio'
The requested action is complete.
```

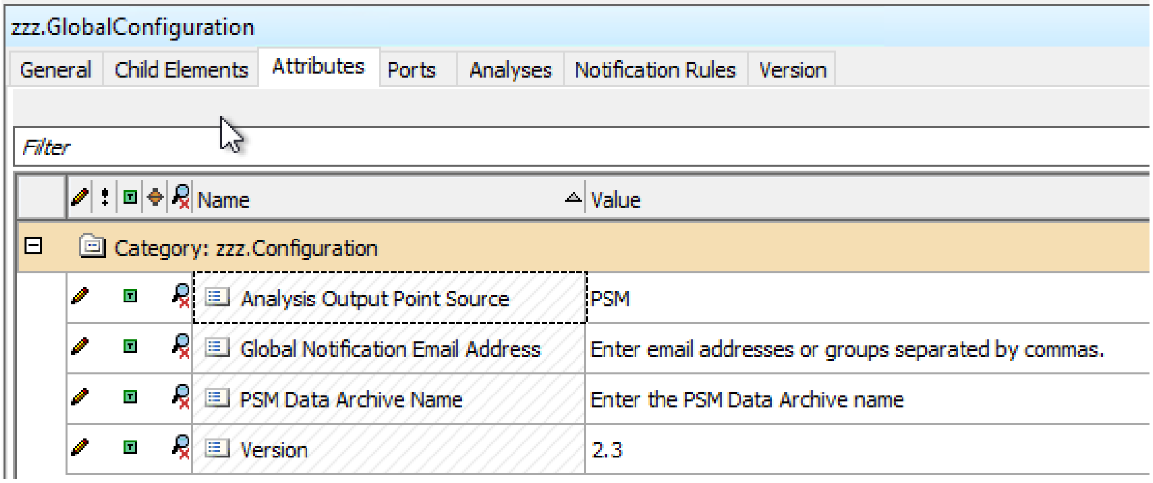
d. Repeat step c. above but select the **PISystemMonitoringExampleKit _vN.N_YYYYMMDD.xml** file to import the PSM example kit. This will take a little bit longer.

i. Once the import is complete, confirm the templates show in the library.

6. Select **Elements** in the navigator to bring up **Elements** in the browser - you will see the element structure in your database – at this time you only have a **zzz.GlobalConfiguration** element which comes with the example kit.



a. Select the element and select the **Attributes** tab in the viewer. You will see there are 4 attributes:



Category: zzz.Configuration

Name	Value
Analysis Output Point Source	PSM
Global Notification Email Address	Enter email addresses or groups separated by commas.
PSM Data Archive Name	Enter the PSM Data Archive name
Version	2.3

Two of these attributes require configuration:

i. Click on the **Value** field next to **Global Notification Email Address** and enter the email address provided in the scenario at the beginning of the course. Ensure it is exactly correct:

student01@pischool.int

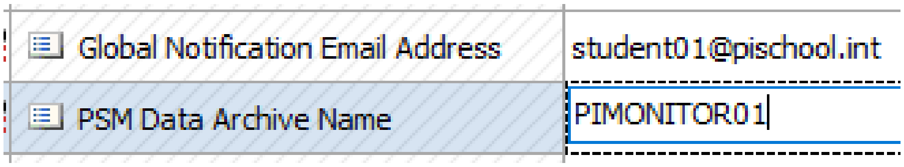
ii. Click on the **Value** field next to **PSM Data Archive Name** and enter the monitoring Data Archive server name provided in the architecture diagram at the beginning of the course. Ensure it is exactly correct:

PIMONITOR01

iii. Click **OK**.

iv. While you can edit the **Analysis Output Point Source**, it has a default value of **PSM** which we recommend leaving unless you have a specific reason to change this.

v. The **Version** is the overall version number for the PSM example kit and is for informational purposes only.

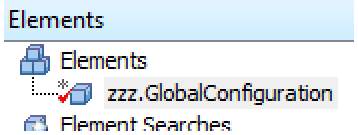


Name	Value
Global Notification Email Address	student01@pischool.int
PSM Data Archive Name	PIMONITOR01

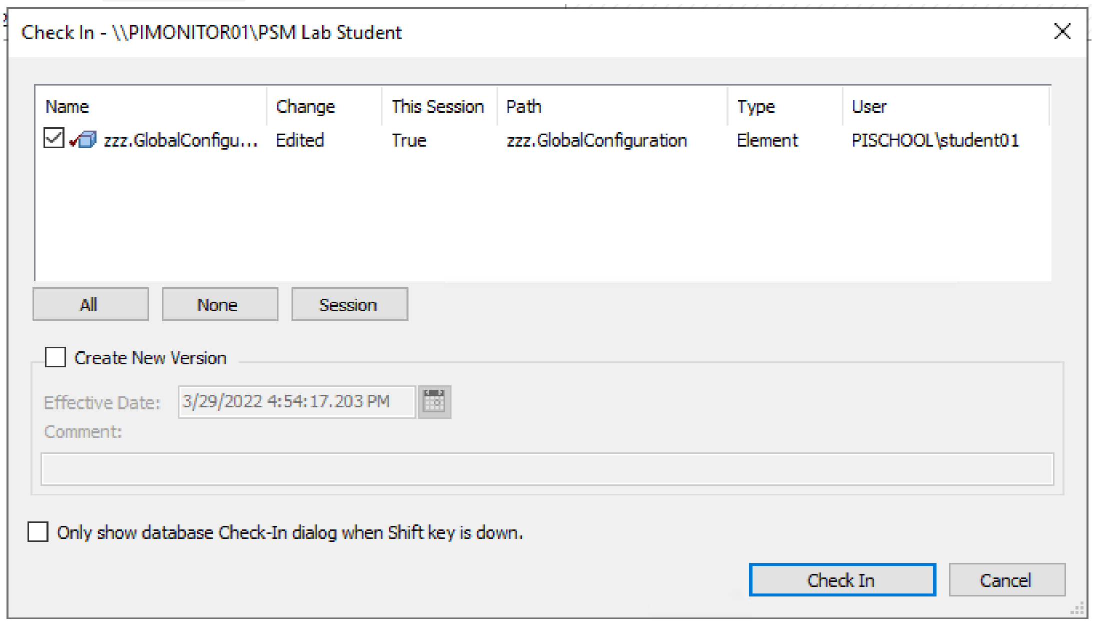
! IMPORTANT	Ensure the values you have entered look EXACTLY as pictured above (no extra spaces or underscores). This is VERY IMPORTANT to set correctly or you will have to delete and recreate all your monitoring tags.

vi. ***NOTE: It is important to get this right before proceeding as the PSM Data Archive Name is used by reference in all tag naming. If this is set wrong and you proceed with tag creation, the tag name references won't work and you will have a mess of tag names to clean up (delete and re-create). This also sets the default email for notifications. It can be overridden at each level if necessary (ex: want to send alerts for a specific server to a site support team), but this will be used by default.***

7. In the browser you will see a red checkmark and star next to the **zzz.GlobalConfiguration** element – this indicates there are changes to this item which have not yet been saved/checked-in.



   a. Click on Check In in the toolbar and click **Check In** on the window to save your changes.



Check In - \\PIMONITOR01\PSM Lab Student

Name	Change	This Session	Path	Type	User
☑ zzz.GlobalConfigu...	Edited	True	zzz.GlobalConfiguration	Element	PISCHOOL\student01

□ Create New Version

Effective Date: 3/29/2022 4:54:17.203 PM

Comment:

□ Only show database Check-In dialog when Shift key is down.

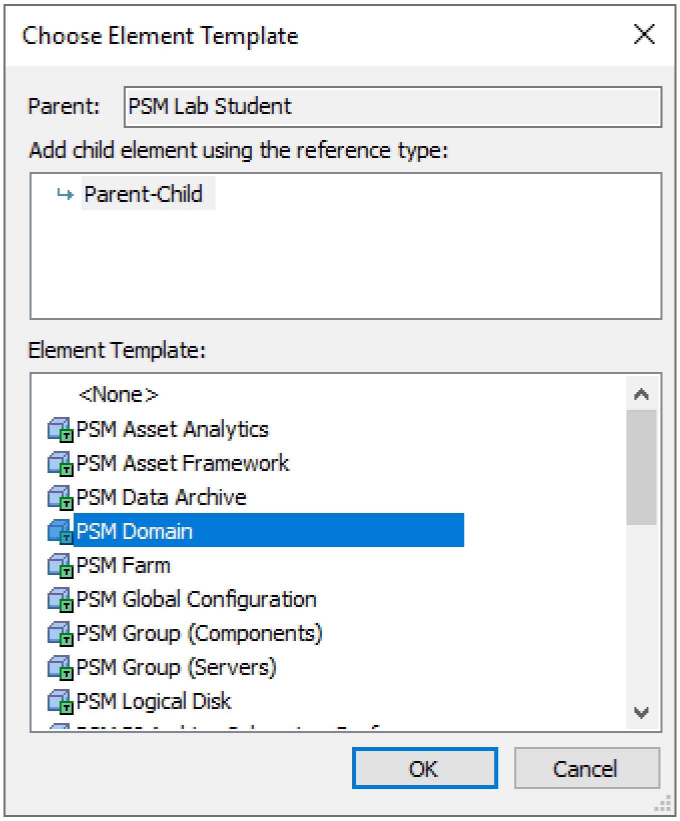
Tip

Changes you make in the AF database have to be checked in to take effect.
If at any time you run into issues where values aren't updating, analyses aren't triggering, or things aren't working as expected, make sure to check in.

Also, you may need to use the Refresh button on the toolbar to see changes take effect.

8. Now you'll proceed with creating your hierarchy. Based on the architecture diagram, all systems to be monitored are on the same network domain, so you will only require a single **Domain** element.

a. Right-click on **Elements** in the browser and select **New Element**. Select the **PSM Domain** template and click **OK**.



Choose Element Template

Parent: PSM Lab Student

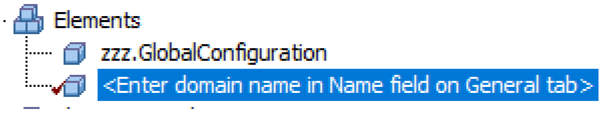
Add child element using the reference type:

Parent-Child

Element Template:

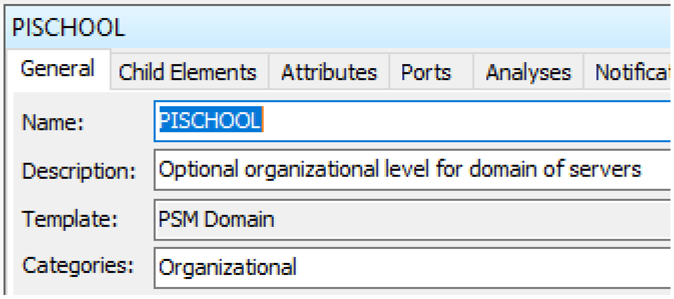
<None>
PSM Asset Analytics
PSM Asset Framework
PSM Data Archive
PSM Domain
PSM Farm
PSM Global Configuration
PSM Group (Components)
PSM Group (Servers)
PSM Logical Disk

b. A **Domain** element will be created with the default name indicating that user input is required:



Elements
zzz.GlobalConfiguration
<Enter domain name in Name field on General tab>

c. On the **General** tab in the viewer change the Name to '**PISCHOOL**'



PISCHOOL

Name: PISCHOOL
Description: Optional organizational level for domain of servers
Template: PSM Domain
Categories: Organizational

d. If you select the **Attributes** tab in the viewer, you will see there are no attributes for this element, apart from the element name (for easy use in PI Vision). It simply acts as an organizational level in the hierarchy.

e. For this course we will only use the **Domain** organizational level, not **Farm** or **Group (*).**

f. Remember to check-in your changes.

5. Exercise 2: Data Archive – failed subsystem

Exercise objectives

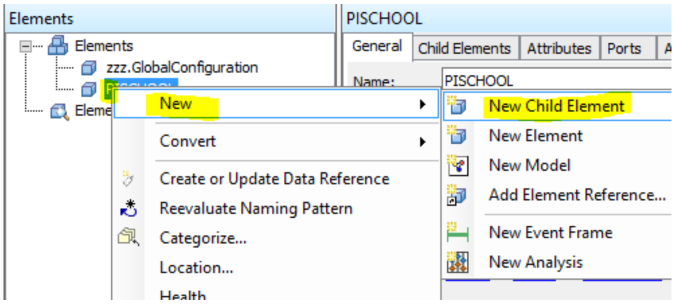
- Create and configure the elements in your AF structure for monitoring a Data Archive
- Create required monitoring tags
- Understand the analyses used for monitoring
- Simulate a failed subsystem condition (you will use backup subsystem)
- Understand an end-to-end use case for PI Data Archive monitoring

Scenario

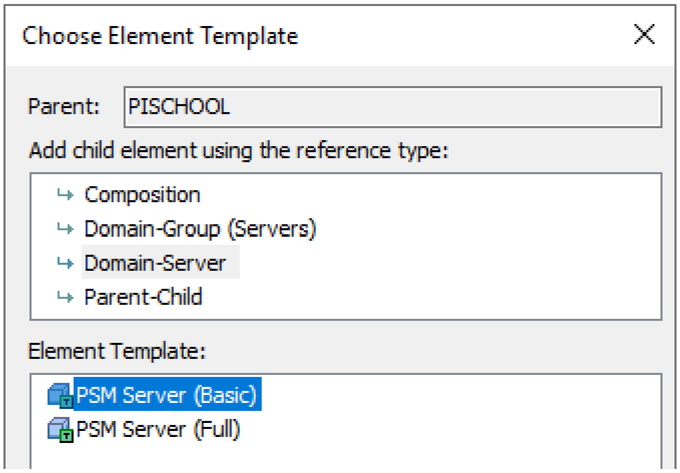
You will review the library templates associated with a PI Data Archive component and note what configuration data is required for them to function. After creating the necessary element, you will locate the required data and configure the element appropriately. You will review the analyses and determine how they are used for monitoring, and what triggers them. You will simulate a failed subsystem condition (using the PI Backup Subsystem) by triggering an analysis to produce an event frame and notification and review these outputs. This will illustrate a full end-to-end use case for PI System monitoring.

Approach

1. Refer the PI architecture diagram provided in the scenario at the beginning of this course manual to get the machine name of the Data Archive.
2. First you will create the element for the Data Archive machine. It uses the **PSM Server (*)** template which can be used for any server, regardless of which PI System components are installed.
   a. In PI System Explorer, in the browser right-click the **PISCHOOL** element in your database and select **New > New Child Element**



b. In the Choose Element Template window, ensure reference type **Domain-Server** is chosen, and select the **PSM Server (Basic)** template:



Choose Element Template

Parent: PISCHOOL

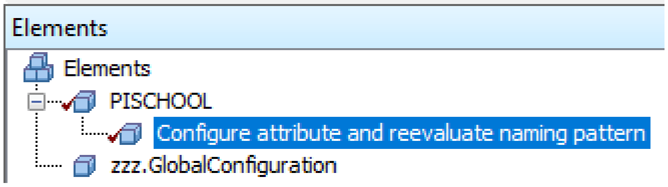
Add child element using the reference type:

- Composition
- Domain-Group (Servers)
- Domain-Server
- Parent-Child

Element Template:

- PSM Server (Basic)
- PSM Server (Full)

c. You will see a new **Server** element under **PISCHOOL** in the element structure. The element will have a default name indicating that user input is required:



Elements

- Elements
  - PISCHOOL
    - Configure attribute and reevaluate naming pattern
  - zzz.GlobalConfiguration

d. Select the **Attributes** tab in the viewer – note that the **PSM Server (Basic)** template only includes configuration attributes and a couple of high level summary and system attributes for general availability. For now we just need a Server element to exist so we can create the Data Archive component under it.

i. The **PSM Server (Full)** template includes several more attributes for detailed server monitoring. If you wish to learn more about the **PSM Server (*)** templates see *Optional exercise 1: Server – high processor utilization*

e. Expand the **Target** attribute (click the + to the left of Target). Here you will need to fill in certain details that were provided in the scenario at the beginning of this document:

i. **Target** (machine name for this server)

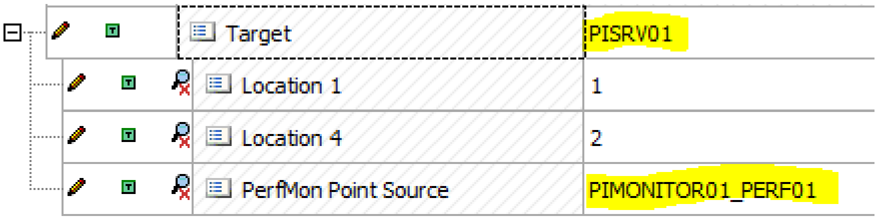
PISRV01

ii. The default values for **Location 1** and **Location 4** can stay as-is

iii. **PerfMon Point Source** (point source for the PI Interface for Performance Monitor (PIPerfMon) instance that is retrieving the performance counters for this server). The PIPerfMon point source is provided in the Scenario at the beginning of the course.

PIMONITOR01_PERF01

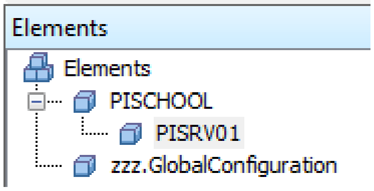
For information on how to find the PIPerfMon point source see *How to find the PIPerfMon point source.*



Target	PISRV01
Location 1	1
Location 4	2
PerfMon Point Source	PIMONITOR01_PERF01

! IMPORTANT	Ensure the values you have entered look **EXACTLY** as pictured above (no extra spaces).

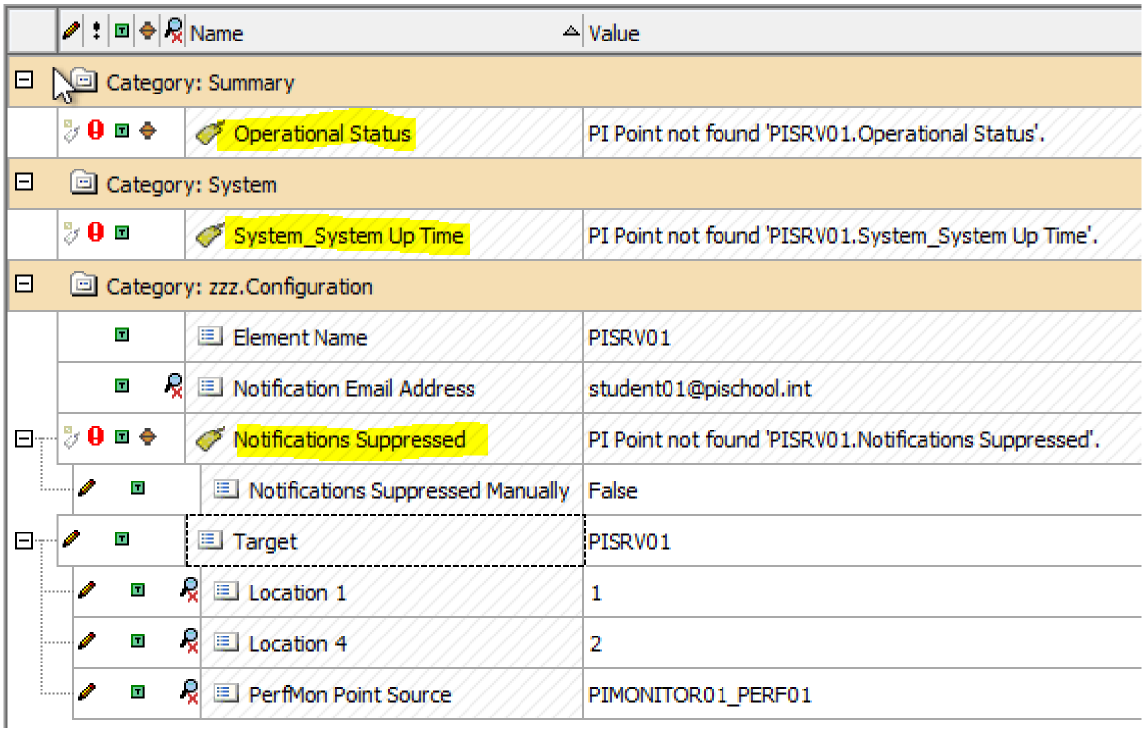
f. Check in your changes. At this point your hierarchy should appear as follows



Elements

- Elements
  - PISCHOOL
    - PISRV01
  - zzz.GlobalConfiguration

g. The **Operational Status**, **System_System Up Time**, and **Notifications Suppressed** attributes are all PI Point data references and the values show 'PI Point not found' – this is because you haven't created the associated PI Tags for them.



Name	Value
Category: Summary	
Operational Status	PI Point not found 'PISRV01.Operational Status'.
Category: System	
System_System Up Time	PI Point not found 'PISRV01.System_System Up Time'.
Category: zzz.Configuration	
Element Name	PISRV01
Notification Email Address	student01@pischool.int
Notifications Suppressed	PI Point not found 'PISRV01.Notifications Suppressed'.
Notifications Suppressed Manually	False
Target	PISRV01
Location 1	1
Location 4	2
PerfMon Point Source	PIMONITOR01_PERF01

i. These attributes are used to monitor the overall availability status of the server, and in certain cases can be used to suppress notifications when a server or component is unavailable. See the following for more information on:

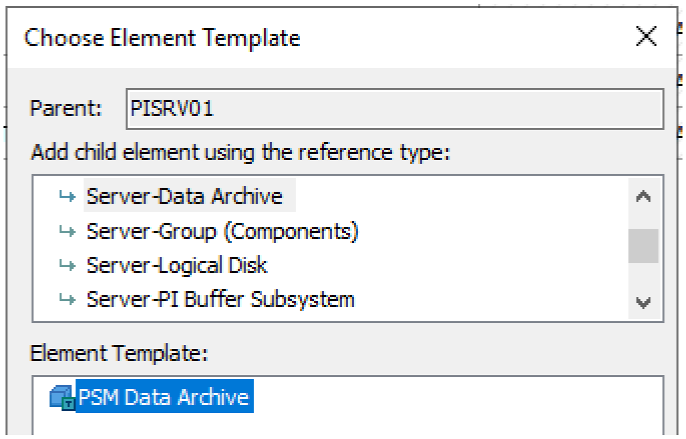
1. server-level monitoring: *Optional exercise 1: Server – high processor utilization*
2. *Notification suppression*

ii. We will create the PI Tags for this in a future step, so there is no further configuration needed right now.

3. Next you will create the element for the Data Archive component. As you will see, there are many aspects of the Data Archive that are important to monitor – things like subsystems, archiving, event queues, backups, out of order events, etc. Monitoring the health of your Data Archive allows you to proactively address issues that arise.

a. In PI System Explorer, in the browser right-click the **PISRV01** element in your database and select **New > New Child Element**.

   i. In the Choose Element Template window, ensure reference type **Server-Data Archive** is chosen, and select the **PSM Data Archive** template:



Choose Element Template

Parent: PISRV01

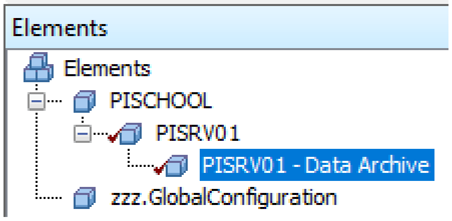
Add child element using the reference type:

- Server-Data Archive
- Server-Group (Components)
- Server-Logical Disk
- Server-PI Buffer Subsystem

Element Template:

PSM Data Archive

   ii. You will see a new **Data Archive** element with a default name under **PISRV01** in the element structure. This name can stay as-is:

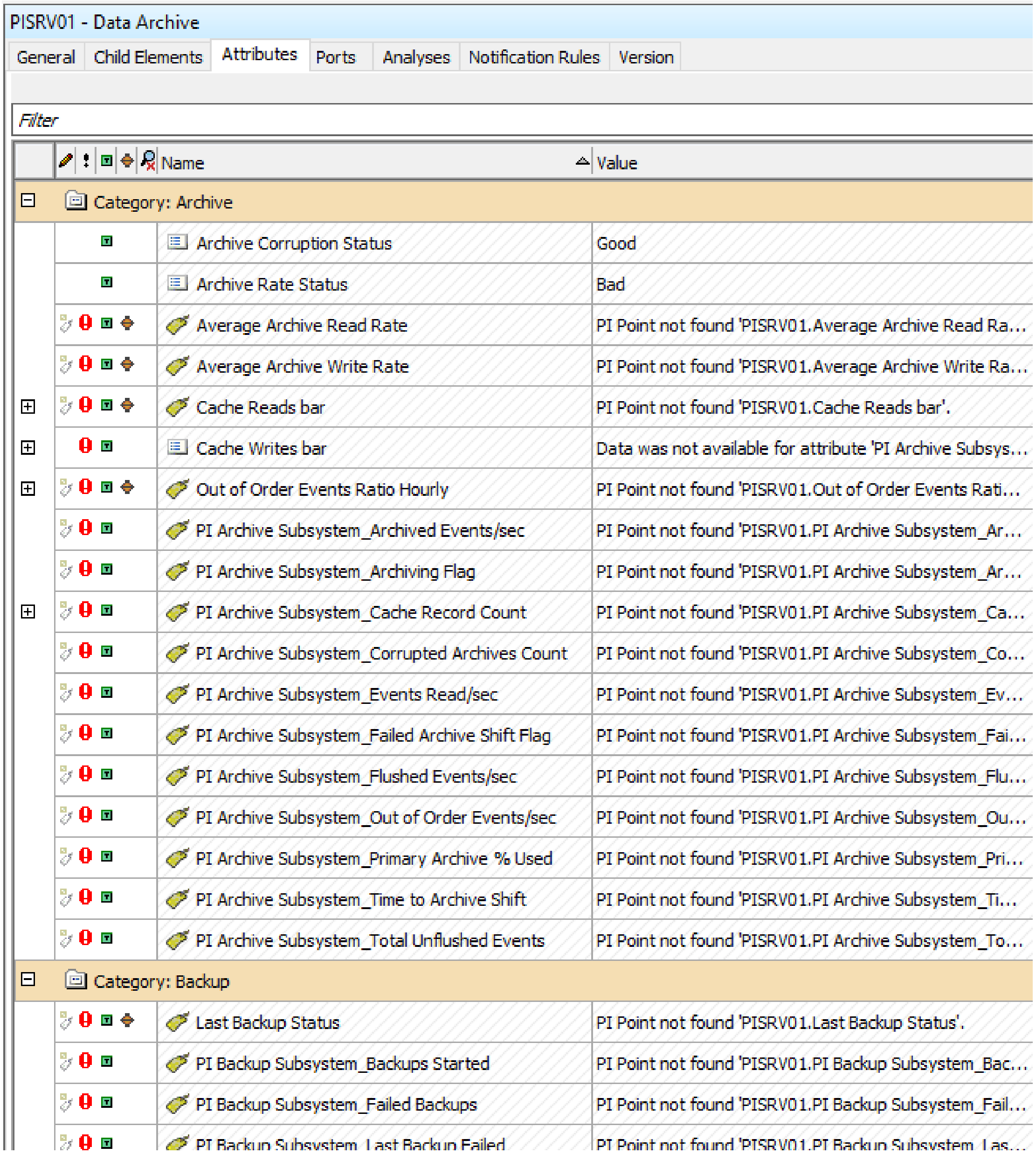


Elements

- Elements
  - PISCHOOL
    - PISRV01
      - PISRV01 - Data Archive
  - zzz.GlobalConfiguration

b. In the Viewer on the **Attributes** tab, you will see all of the attributes defined for this element. Notice there are many attributes which have PI Point data references, and that the values show

'PI Point not found' – this is because you haven't created the associated PI Tags yet.

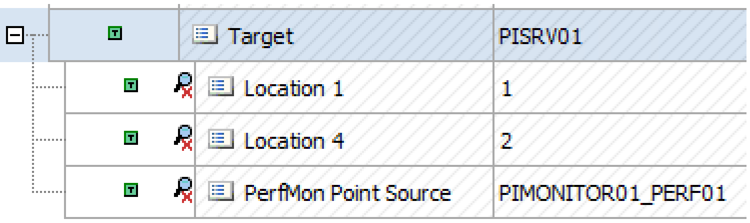


PISRV01 - Data Archive

Name	Value
Category: Archive	
Archive Corruption Status	Good
Archive Rate Status	Bad
Average Archive Read Rate	PI Point not found 'PISRV01.Average Archive Read Ra…
Average Archive Write Rate	PI Point not found 'PISRV01.Average Archive Write Ra…
Cache Reads bar	PI Point not found 'PISRV01.Cache Reads bar'.
Cache Writes bar	Data was not available for attribute 'PI Archive Subsys…
Out of Order Events Ratio Hourly	PI Point not found 'PISRV01.Out of Order Events Rati…
PI Archive Subsystem_Archived Events/sec	PI Point not found 'PISRV01.PI Archive Subsystem_Ar…
PI Archive Subsystem_Archiving Flag	PI Point not found 'PISRV01.PI Archive Subsystem_Ar…
PI Archive Subsystem_Cache Record Count	PI Point not found 'PISRV01.PI Archive Subsystem_Ca…
PI Archive Subsystem_Corrupted Archives Count	PI Point not found 'PISRV01.PI Archive Subsystem_Co…
PI Archive Subsystem_Events Read/sec	PI Point not found 'PISRV01.PI Archive Subsystem_Ev…
PI Archive Subsystem_Failed Archive Shift Flag	PI Point not found 'PISRV01.PI Archive Subsystem_Fai…
PI Archive Subsystem_Flushed Events/sec	PI Point not found 'PISRV01.PI Archive Subsystem_Flu…
PI Archive Subsystem_Out of Order Events/sec	PI Point not found 'PISRV01.PI Archive Subsystem_Ou…
PI Archive Subsystem_Primary Archive % Used	PI Point not found 'PISRV01.PI Archive Subsystem_Pri…
PI Archive Subsystem_Time to Archive Shift	PI Point not found 'PISRV01.PI Archive Subsystem_Ti…
PI Archive Subsystem_Total Unflushed Events	PI Point not found 'PISRV01.PI Archive Subsystem_To…
Category: Backup	
Last Backup Status	PI Point not found 'PISRV01.Last Backup Status'.
PI Backup Subsystem_Backups Started	PI Point not found 'PISRV01.PI Backup Subsystem_Bac…
PI Backup Subsystem_Failed Backups	PI Point not found 'PISRV01.PI Backup Subsystem_Fail…
PI Backup Subsystem_Last Backup Failed	PI Point not found 'PISRV01.PI Backup Subsystem_Las…

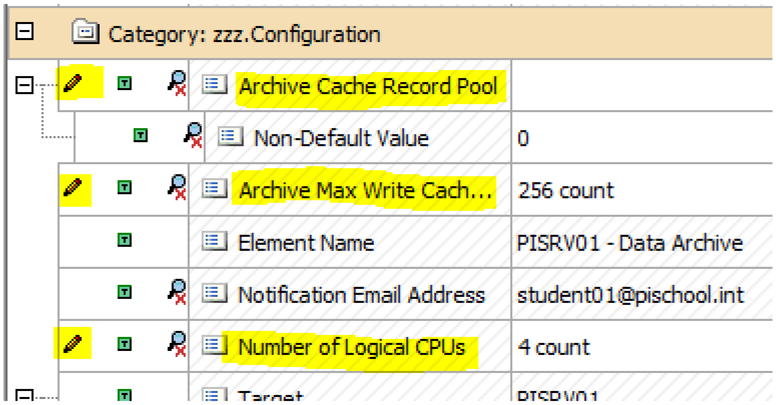
! IMPORTANT	Ensure you are seeing 'PI Point not found' in the value for the attributes. If you are seeing any other messages including 'Data Archive…not found', please reach out to the Learning Support Team

i. Generally the items in the zzz.Configuration category are either inherited from a parent (e.g., Target, Notification Email Address) or need to be set appropriately by the user prior to creating monitoring tags (as some of them are used in PI Tag configuration). Find and expand the **Target** attribute. If you scroll down it should be at the bottom under **Category: zzz.Configuration**. Notice that the **Target** and **PerfMon Point Source** attributes are already filled in – they are inherited from the parent element. Ensure yours match what is shown below.



Target	PISRV01
Location 1	1
Location 4	2
PerfMon Point Source	PIMONITOR01_PERF01

ii. Look for other items under zzz.Configuration that have the ✎ pencil icon to the left – these likely need to be set by the user. Where possible attributes have default values but you should confirm those values meet your needs.

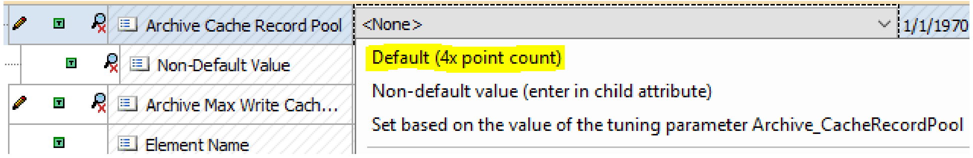


Category: zzz.Configuration

Archive Cache Record Pool	
Non-Default Value	0
Archive Max Write Cach...	256 count
Element Name	PISRV01 - Data Archive
Notification Email Address	student01@pischool.int
Number of Logical CPUs	4 count
Target	PISRV01

For the Data Archive, you will need to set:

1. **Archive Cache Record Pool** – if you click the down arrow to the right of the blank value field, you will get a dropdown with pre-configured values. For the course (and in most cases) you can select **Default (4x point count)**.

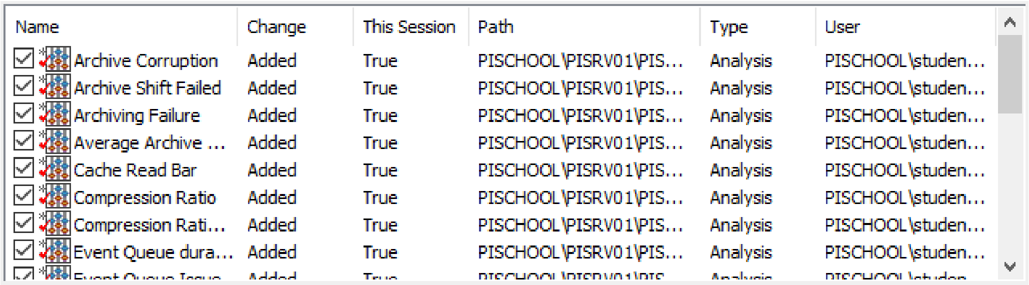


Archive Cache Record Pool	<None>
Non-Default Value	
Archive Max Write Cach...	
Element Name	

1/1/1970
Default (4x point count)
Non-default value (enter in child attribute)
Set based on the value of the tuning parameter Archive_CacheRecordPool

2. **Archive Max Write Cache Per Point** – for the course (and in most cases) the default of 256 count is good
3. **Number of Logical CPUs** – set this appropriately for your system. For the course, 4 is correct.

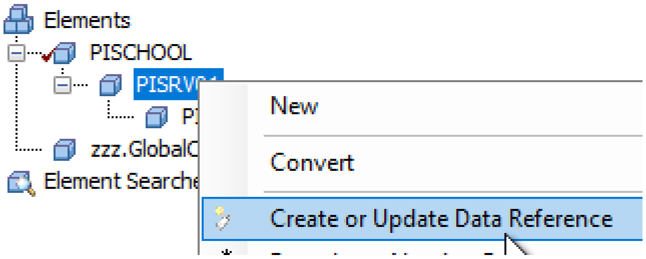
iii. Check in your changes. You will notice a large list of added/changed items (analyses, elements) that are created with this type of element.



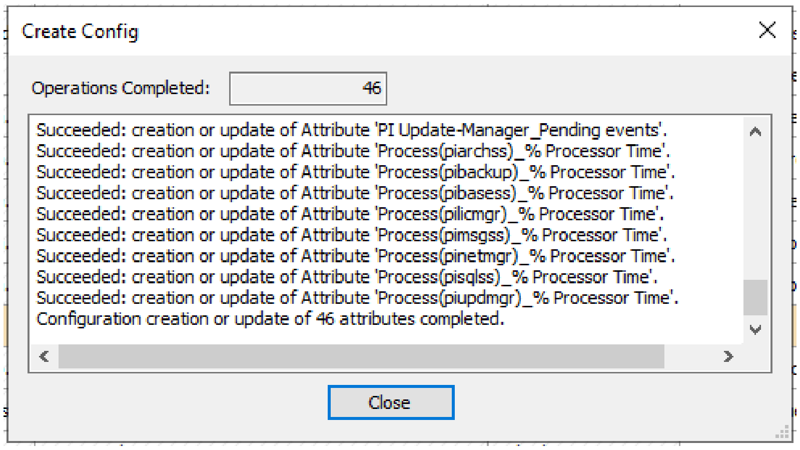
Name	Change	This Session	Path	Type	User
☑ Archive Corruption	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Archive Shift Failed	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Archiving Failure	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Average Archive ...	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Cache Read Bar	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Compression Ratio	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Compression Rati...	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Event Queue dura...	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...
☑ Event Queue Issue	Added	True	PISCHOOL\PISRV01\PIS...	Analysis	PISCHOOL\studen...

iv. You will now create the PI Tags required for both the Server and Data Archive components. Right-click on the **PISRV01** element in the browser and select **Create or**

Update Data Reference.



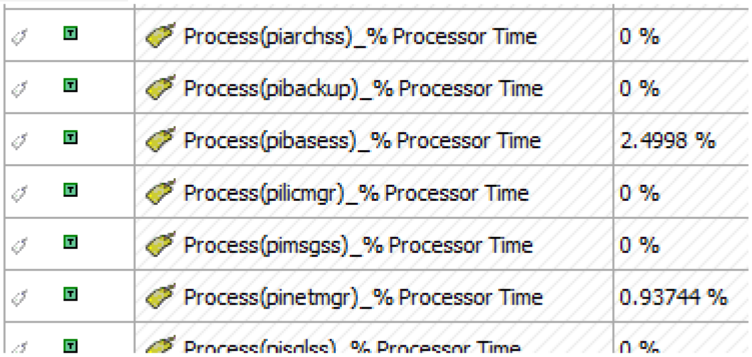
v. PI Tags will be created for both the PISRV01 and PISRV01 – Data Archive elements. The Create Config window will show the progress of the creation.



Operations Completed: 46

```
Succeeded: creation or update of Attribute 'PI Update-Manager_Pending events'.
Succeeded: creation or update of Attribute 'Process(piarchss)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pibackup)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pibasess)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pilicmgr)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pimsgss)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pinetmgr)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(pisqlss)_% Processor Time'.
Succeeded: creation or update of Attribute 'Process(piupdmgr)_% Processor Time'.
Configuration creation or update of 46 attributes completed.
```

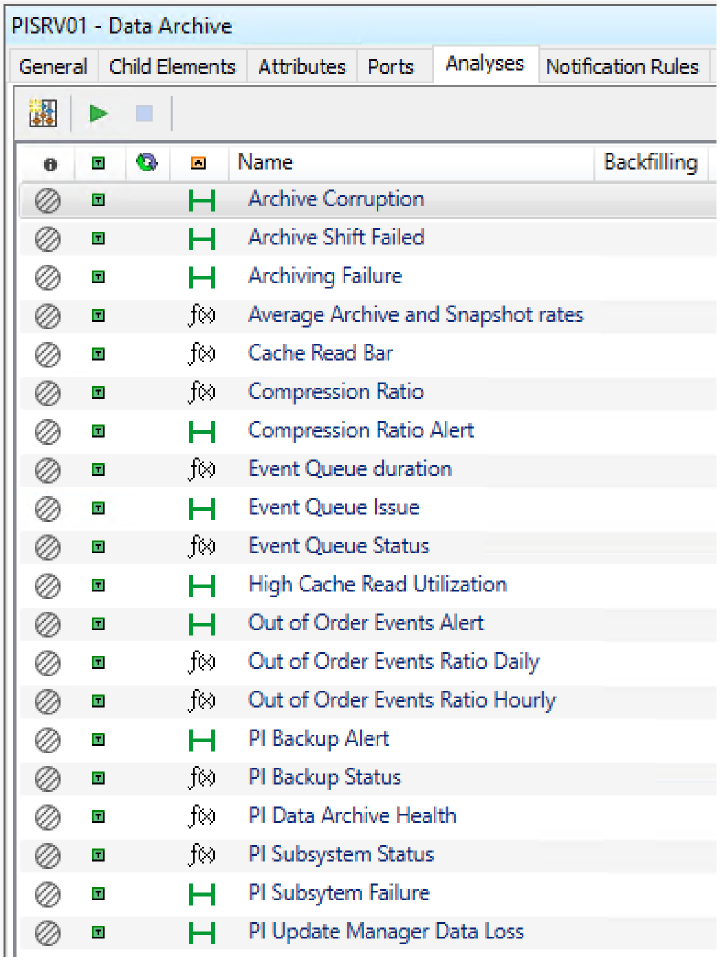
vi. After it is complete you will see a value of **Pt Created** beside the attributes that have PI Point data references. After a couple of minutes the PI Tags will be picked up by PIPerfMon, and will start showing values in PI System Explorer. You may need to click Refresh on the toolbar.



▣	Process(piarchss)_% Processor Time	0 %
▣	Process(pibackup)_% Processor Time	0 %
▣	Process(pibasess)_% Processor Time	2.4998 %
▣	Process(pilicmgr)_% Processor Time	0 %
▣	Process(pimsgss)_% Processor Time	0 %
▣	Process(pinetmgr)_% Processor Time	0.93744 %
▣	Process(pisqlss)_% Processor Time	0 %

! IMPORTANT	Ensure you are seeing values for the PI Point data reference attributes (anything with only these 2 icons to the left: ▣ 🏷). Other types of attributes may or may not have values, including staying as 'Pt Created'. If you are not seeing values, or are still seeing 'Pt Created' on the PI Point data reference attributes after a few minutes, please reach out to the Learning Support Team.

c. On the **PISRV01 – Data Archive** element, select the **Analyses** tab in the viewer and you will see many analyses defined for this element. The ⊢⊣ icon indicates event frames, meaning if the trigger conditions are reached an event frame will be created. The f(x) icon indicates expressions which are calculations. Notice the ▨ icon to the left of each of the analyses – this means they are currently disabled.

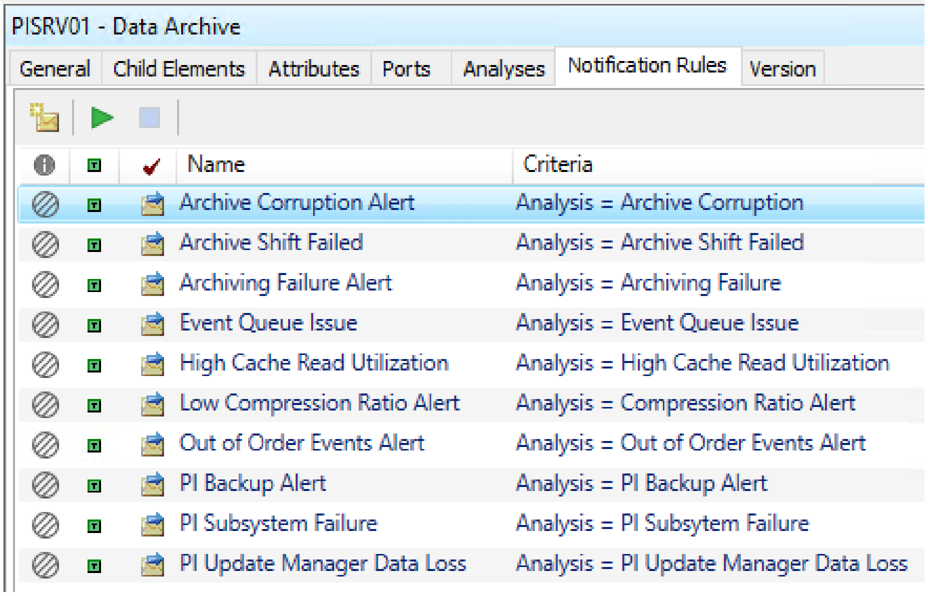


PISRV01 - Data Archive

General | Child Elements | Attributes | Ports | Analyses | Notification Rules

		Name	Backfilling
▨	H	Archive Corruption	
▨	H	Archive Shift Failed	
▨	H	Archiving Failure	
▨	f(x)	Average Archive and Snapshot rates	
▨	f(x)	Cache Read Bar	
▨	f(x)	Compression Ratio	
▨	H	Compression Ratio Alert	
▨	f(x)	Event Queue duration	
▨	H	Event Queue Issue	
▨	f(x)	Event Queue Status	
▨	H	High Cache Read Utilization	
▨	H	Out of Order Events Alert	
▨	f(x)	Out of Order Events Ratio Daily	
▨	f(x)	Out of Order Events Ratio Hourly	
▨	H	PI Backup Alert	
▨	f(x)	PI Backup Status	
▨	f(x)	PI Data Archive Health	
▨	f(x)	PI Subsystem Status	
▨	H	PI Subsytem Failure	
▨	H	PI Update Manager Data Loss	

d. Select the **Notification Rules** tab in the viewer and you will see several notification rules defined for this element as well. Note they are also disabled.

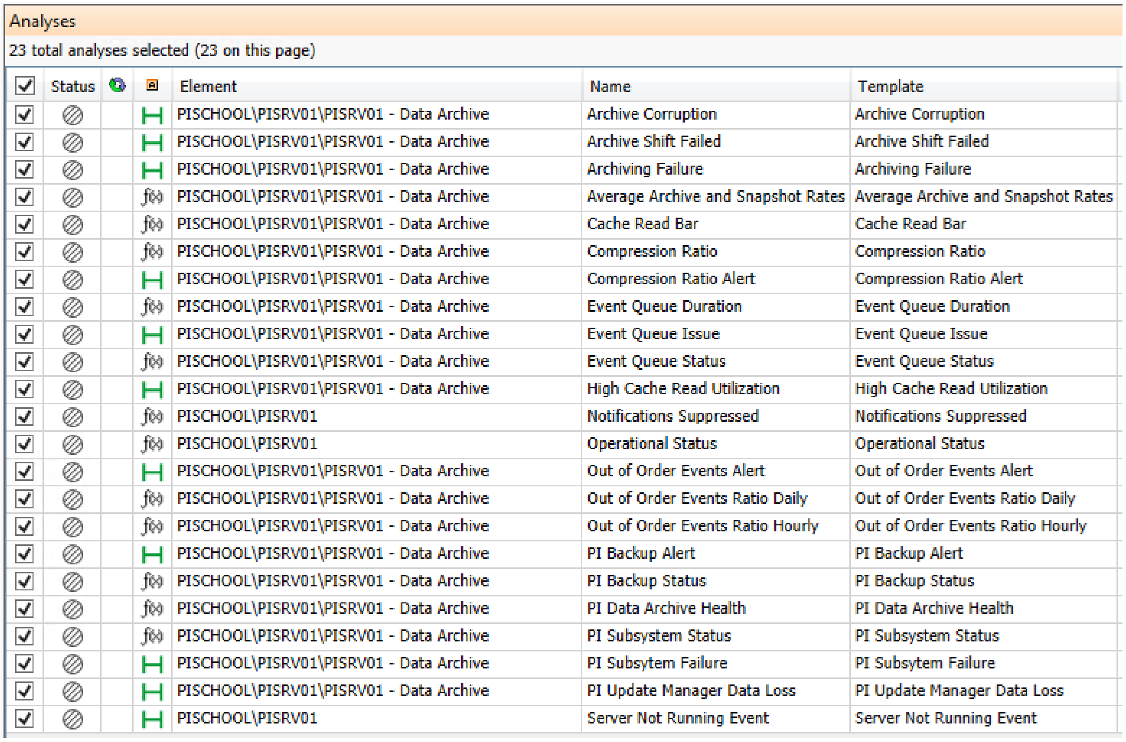


PISRV01 - Data Archive

General | Child Elements | Attributes | Ports | Analyses | Notification Rules | Version

	Name	Criteria
▨	Archive Corruption Alert	Analysis = Archive Corruption
▨	Archive Shift Failed	Analysis = Archive Shift Failed
▨	Archiving Failure Alert	Analysis = Archiving Failure
▨	Event Queue Issue	Analysis = Event Queue Issue
▨	High Cache Read Utilization	Analysis = High Cache Read Utilization
▨	Low Compression Ratio Alert	Analysis = Compression Ratio Alert
▨	Out of Order Events Alert	Analysis = Out of Order Events Alert
▨	PI Backup Alert	Analysis = PI Backup Alert
▨	PI Subsystem Failure	Analysis = PI Subsytem Failure
▨	PI Update Manager Data Loss	Analysis = PI Update Manager Data Loss

e. To enable the analyses and notification rules select **Management** in the navigator. In the browser, ensure **Analyses** is selected under Type and **All** is selected under Analysis Searches. In the viewer you will see a list of the analyses - click the empty checkbox above the list to select them all including the three associated with the server template.

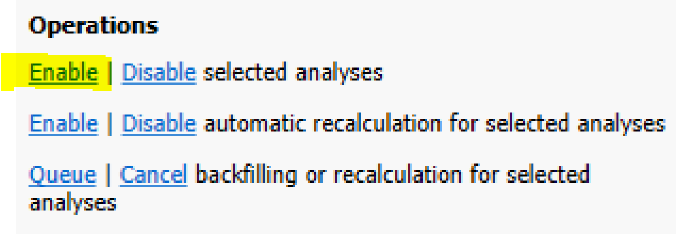


Analyses

23 total analyses selected (23 on this page)

☑	Status			Element	Name	Template
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archive Corruption	Archive Corruption
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archive Shift Failed	Archive Shift Failed
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archiving Failure	Archiving Failure
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Average Archive and Snapshot Rates	Average Archive and Snapshot Rates
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Cache Read Bar	Cache Read Bar
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Compression Ratio	Compression Ratio
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Compression Ratio Alert	Compression Ratio Alert
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Event Queue Duration	Event Queue Duration
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Event Queue Issue	Event Queue Issue
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Event Queue Status	Event Queue Status
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	High Cache Read Utilization	High Cache Read Utilization
☑	⊘		f(x)	PISCHOOL\PISRV01	Notifications Suppressed	Notifications Suppressed
☑	⊘		f(x)	PISCHOOL\PISRV01	Operational Status	Operational Status
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	Out of Order Events Alert	Out of Order Events Alert
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Out of Order Events Ratio Daily	Out of Order Events Ratio Daily
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	Out of Order Events Ratio Hourly	Out of Order Events Ratio Hourly
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Backup Alert	PI Backup Alert
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Backup Status	PI Backup Status
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Data Archive Health	PI Data Archive Health
☑	⊘		f(x)	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Subsystem Status	PI Subsystem Status
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Subsytem Failure	PI Subsytem Failure
☑	⊘		H	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Update Manager Data Loss	PI Update Manager Data Loss
☑	⊘		H	PISCHOOL\PISRV01	Server Not Running Event	Server Not Running Event

i. On the right pane select **Enable** to enable the analyses.



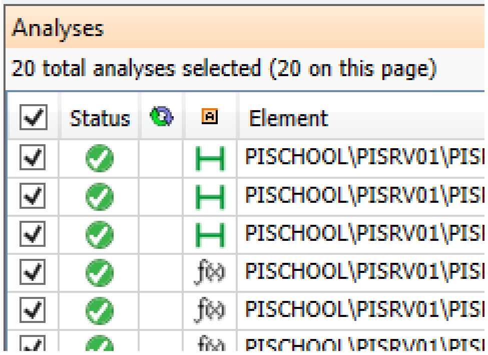
Operations

Enable | Disable selected analyses

Enable | Disable automatic recalculation for selected analyses

Queue | Cancel backfilling or recalculation for selected analyses

ii. The status icon will change to ○ while the analyses are starting and then show as below once they are running:

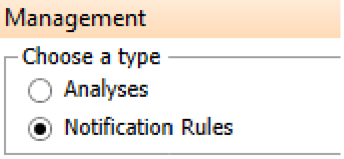


Analyses

20 total analyses selected (20 on this page)

☑	Status			Element
☑	✓		H	PISCHOOL\PISRV01\PISI
☑	✓		H	PISCHOOL\PISRV01\PISI
☑	✓		H	PISCHOOL\PISRV01\PISI
☑	✓		f(x)	PISCHOOL\PISRV01\PISI
☑	✓		f(x)	PISCHOOL\PISRV01\PISI
☑	✓		f(x)	PISCHOOL\PISRV01\PISI

iii. The notification rules also need to be enabled – select **Notification Rules** under Type on the top left:



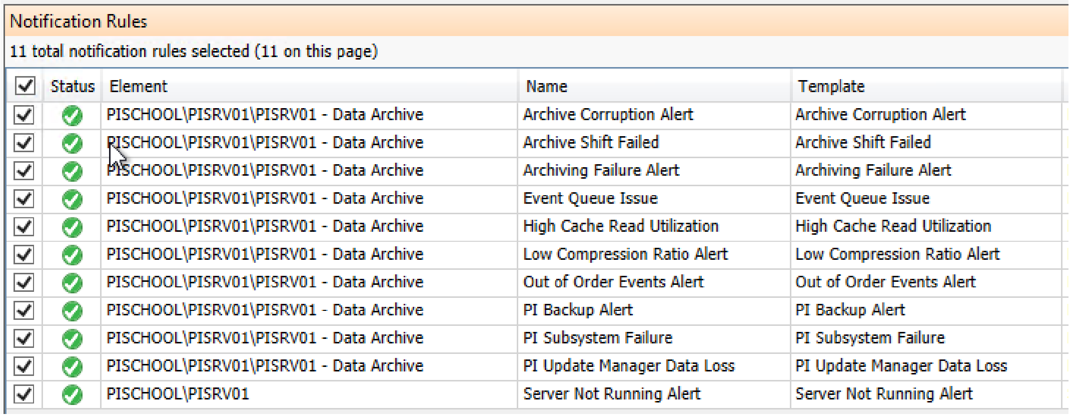
Management

Choose a type

○ Analyses

● Notification Rules

iv. Select all the notification rules, including the one for the server template, and enable them.



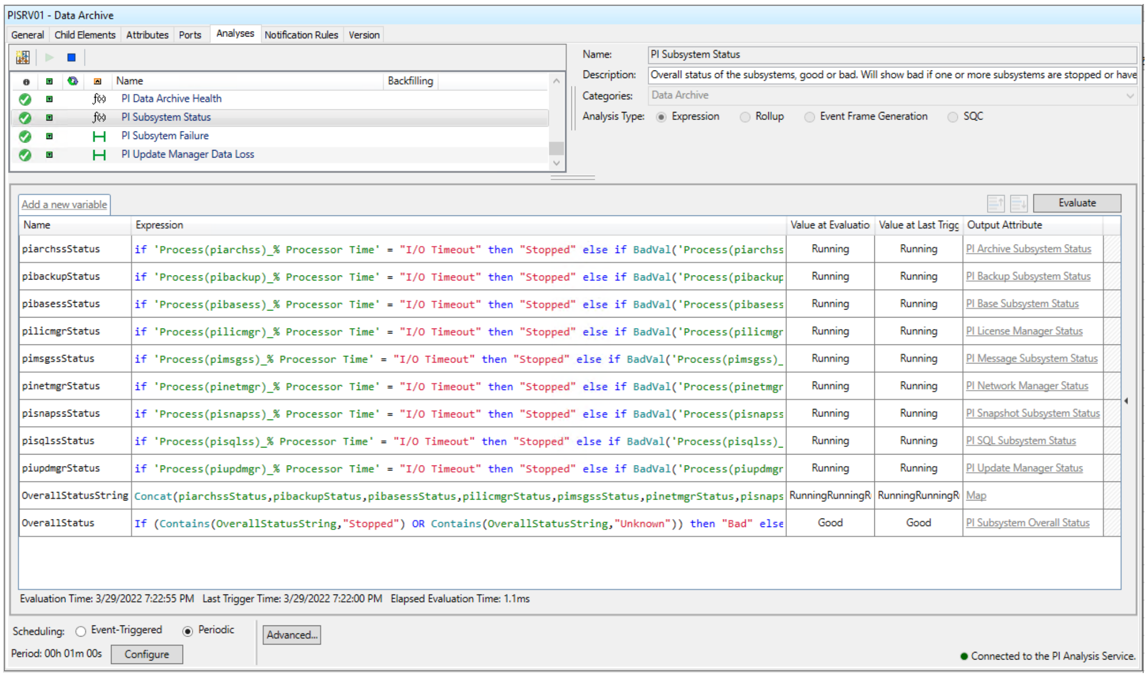
Notification Rules

11 total notification rules selected (11 on this page)

☑	Status	Element	Name	Template
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archive Corruption Alert	Archive Corruption Alert
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archive Shift Failed	Archive Shift Failed
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Archiving Failure Alert	Archiving Failure Alert
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Event Queue Issue	Event Queue Issue
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	High Cache Read Utilization	High Cache Read Utilization
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Low Compression Ratio Alert	Low Compression Ratio Alert
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	Out of Order Events Alert	Out of Order Events Alert
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Backup Alert	PI Backup Alert
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Subsystem Failure	PI Subsystem Failure
☑	✓	PISCHOOL\PISRV01\PISRV01 - Data Archive	PI Update Manager Data Loss	PI Update Manager Data Loss
☑	✓	PISCHOOL\PISRV01	Server Not Running Alert	Server Not Running Alert

4. Now that you have created and configured the element and analyses, you will simulate a failed subsystem condition using the PI Backup Subsystem. The Data Archive has several subsystems that work together to make the overall Data Archive function properly. If any one of these subsystems fails it can cause problems with the Data Archive. It's important to know when a subsystem has failed so it can be addressed quickly to prevent additional issues or extended system down time.

   a. Select **Elements** in the navigator and within the **PISRV01 – Data Archive** element go to the **Analyses** tab in the viewer and select the **PI Subsystem Status** expression. In the lower window you will see multiple variables defined:



PISRV01 - Data Archive

General | Child Elements | Attributes | Ports | Analyses | Notification Rules | Version

Name	Backfilling
PI Data Archive Health	
PI Subsystem Status	
PI Subsytem Failure	
PI Update Manager Data Loss	

Name: PI Subsystem Status

Description: Overall status of the subsystems, good or bad. Will show bad if one or more subsystems are stopped or have

Categories: Data Archive

Analysis Type: ● Expression ○ Rollup ○ Event Frame Generation ○ SQC

Add a new variable

Name	Expression	Value at Evaluatio	Value at Last Trigg	Output Attribute
piarchssStatus	`if 'Process(piarchss)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(piarchss`	Running	Running	PI Archive Subsystem Status
pibackupStatus	`if 'Process(pibackup)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pibackup`	Running	Running	PI Backup Subsystem Status
pibasessStatus	`if 'Process(pibasess)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pibasess`	Running	Running	PI Base Subsystem Status
pilicmgrStatus	`if 'Process(pilicmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pilicmgr`	Running	Running	PI License Manager Status
pimsgssStatus	`if 'Process(pimsgss)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pimsgss)_`	Running	Running	PI Message Subsystem Status
pinetmgrStatus	`if 'Process(pinetmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pinetmgr`	Running	Running	PI Network Manager Status
pisnapssStatus	`if 'Process(pisnapss)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pisnapss`	Running	Running	PI Snapshot Subsystem Status
pisqlssStatus	`if 'Process(pisqlss)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(pisqlss)_`	Running	Running	PI SQL Subsystem Status
piupdmgrStatus	`if 'Process(piupdmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(piupdmgr`	Running	Running	PI Update Manager Status
OverallStatusString	`Concat(piarchssStatus,pibackupStatus,pibasessStatus,pilicmgrStatus,pimsgssStatus,pinetmgrStatus,pisnaps`	RunningRunningR	RunningRunningR	Map
OverallStatus	`If (Contains(OverallStatusString,"Stopped") OR Contains(OverallStatusString,"Unknown")) then "Bad" else`	Good	Good	PI Subsystem Overall Status

Evaluation Time: 3/29/2022 7:22:55 PM Last Trigger Time: 3/29/2022 7:22:00 PM Elapsed Evaluation Time: 1.1ms

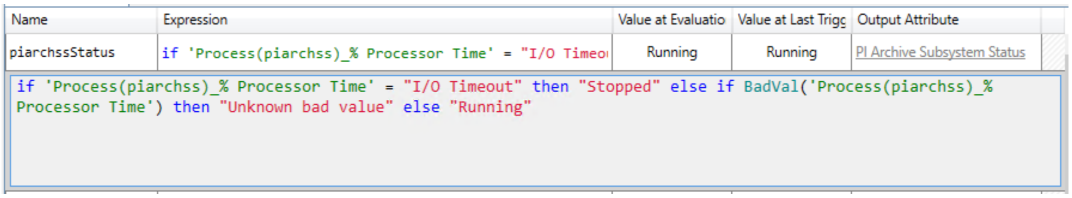
Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 01m 00s

Connected to the PI Analysis Service.

This expression is checking if any of the subsystems are having a problem:

i. Variables for each subsystem which check the value of the **% Processor Time** attribute for that process

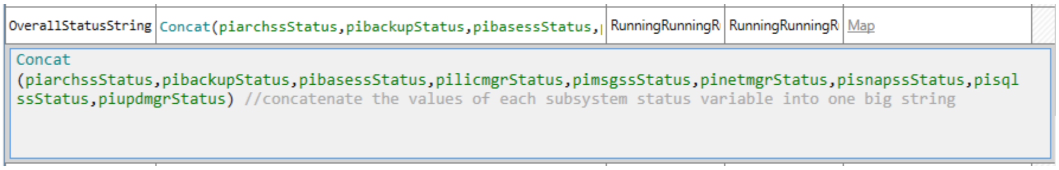


Name	Expression	Value at Evaluatio	Value at Last Trigg	Output Attribute
piarchssStatus	`if 'Process(piarchss)_% Processor Time' = "I/O Timeo`	Running	Running	PI Archive Subsystem Status

```
if 'Process(piarchss)_% Processor Time' = "I/O Timeout" then "Stopped" else if BadVal('Process(piarchss)_%
Processor Time') then "Unknown bad value" else "Running"
```

1. If the attribute value is **I/O Timeout**, the variable value is **Stopped.** (NOTE: I/O Timeout can happen when the PIPerfMon interface is unable to get data from the Windows performance counters.)
2. If the attribute value is bad, the variable value is **Unknown bad value**. (NOTE: BadVal will return **True** if the value does not match the data type – i.e., a system digital state returned for a numeric value – so this captures other states besides I/O Timeout.)
3. For all other attribute values (i.e., good numeric values), the variable value is **Running.**

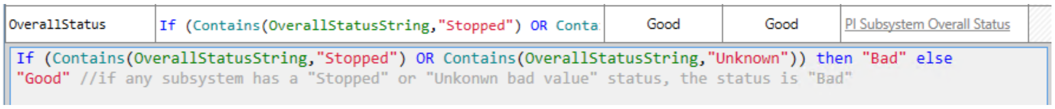
ii. Variable **OverallStatusString** concatenates the values of all the above variables together. If none of the subsystems has a problem, the combined string will only contain multiple instances of the value **Running**.



OverallStatusString	`Concat(piarchssStatus,pibackupStatus,pibasessStatus,`	RunningRunningR	RunningRunningR	Map

```
Concat
(piarchssStatus,pibackupStatus,pibasessStatus,pilicmgrStatus,pimsgssStatus,pinetmgrStatus,pisnapssStatus,pisql
ssStatus,piupdmgrStatus) //concatenate the values of each subsystem status variable into one big string
```

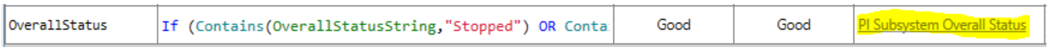
iii. Variable **OverallStatus** checks if the combined value string contains the words **Stopped** or **Unknown** and returns a value of **Bad** if it does. Otherwise it returns **Good**.



OverallStatus	If (Contains(OverallStatusString,"Stopped") OR Conta	Good	Good	PI Subsystem Overall Status

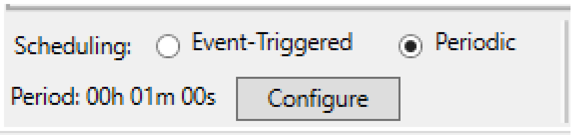
```
If (Contains(OverallStatusString,"Stopped") OR Contains(OverallStatusString,"Unknown")) then "Bad" else
"Good" //if any subsystem has a "Stopped" or "Unkonwn bad value" status, the status is "Bad"
```

iv. The result is captured in the **PI Subsystem Overall Status** output attribute



OverallStatus	If (Contains(OverallStatusString,"Stopped") OR Conta	Good	Good	PI Subsystem Overall Status

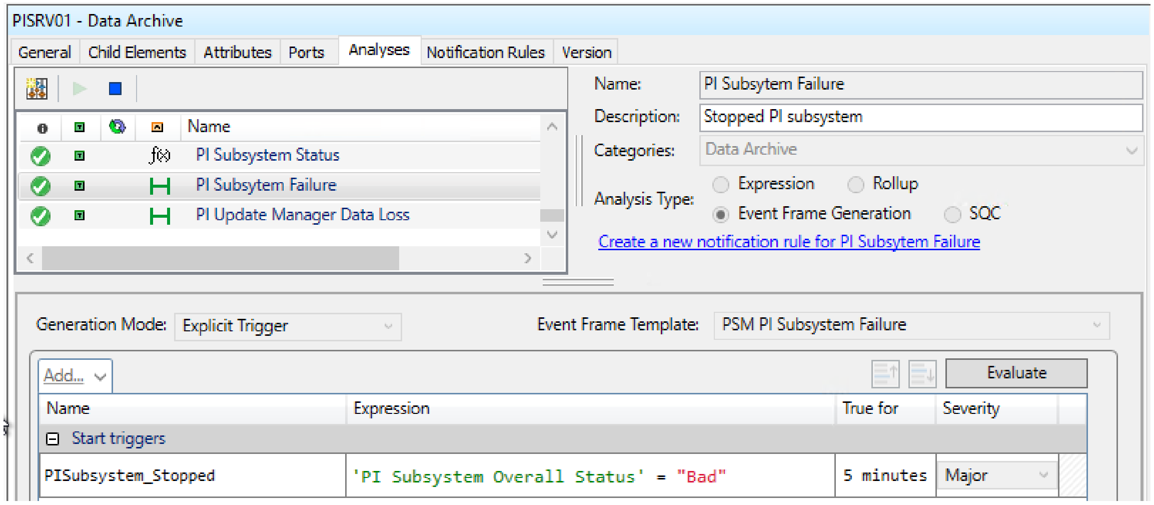
v. The expression is periodic and evaluates every 1 minute



Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 01m 00s

b. Select the **PI Subsystem Failure** event frame. There is one start trigger defined for this event frame, listed in the lower window:



PISRV01 - Data Archive

Name
PI Subsystem Status
PI Subsytem Failure
PI Update Manager Data Loss

Name: PI Subsystem Failure

Description: Stopped PI subsystem

Categories: Data Archive

Analysis Type: ○ Expression ○ Rollup ● Event Frame Generation ○ SQC

Create a new notification rule for PI Subsytem Failure

Generation Mode: Explicit Trigger

Event Frame Template: PSM PI Subsystem Failure

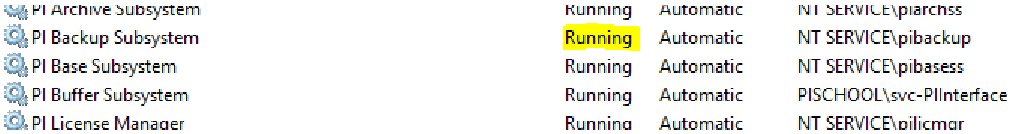
Name	Expression	True for	Severity
⊟ Start triggers			
PISubsystem_Stopped	'PI Subsystem Overall Status' = "Bad"	5 minutes	Major

i. **PISubsystem_Stopped** – checks if the PI Subsystem Overall Status variable from the expression is **Bad**, and triggers if it maintains this value for 5 mins.

ii. The analysis is event triggered so it will check conditions any time the value for the variable changes.

c. To simulate a subsystem failure, you will stop a subsystem on the Data Archive. Connect to PISRV01 using the web browser.

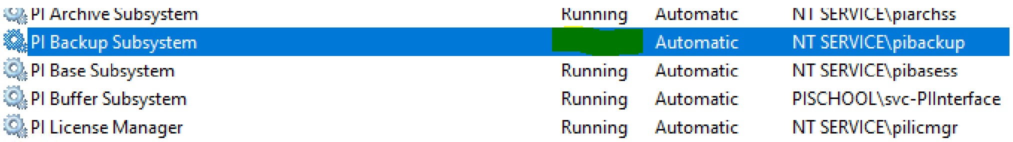
i. On PISRV01, open the Windows **Services** pane via the icon on the taskbar ⚙. Find the **PI Backup Subsystem** and note its current status – it should be **Running**.



PI Archive Subsystem	Running	Automatic	NT SERVICE\piarchss
PI Backup Subsystem	Running	Automatic	NT SERVICE\pibackup
PI Base Subsystem	Running	Automatic	NT SERVICE\pibasess
PI Buffer Subsystem	Running	Automatic	PISCHOOL\svc-PIInterface
PI License Manager	Running	Automatic	NT SERVICE\pilicmgr

ii. Right-click on the service and select **Properties**. Set Startup type to **Disabled** and click **Apply**. (We have a script running on the course machines which catches any PI-related service failures and restarts them automatically, so we need to disable the service so it can't restart.)

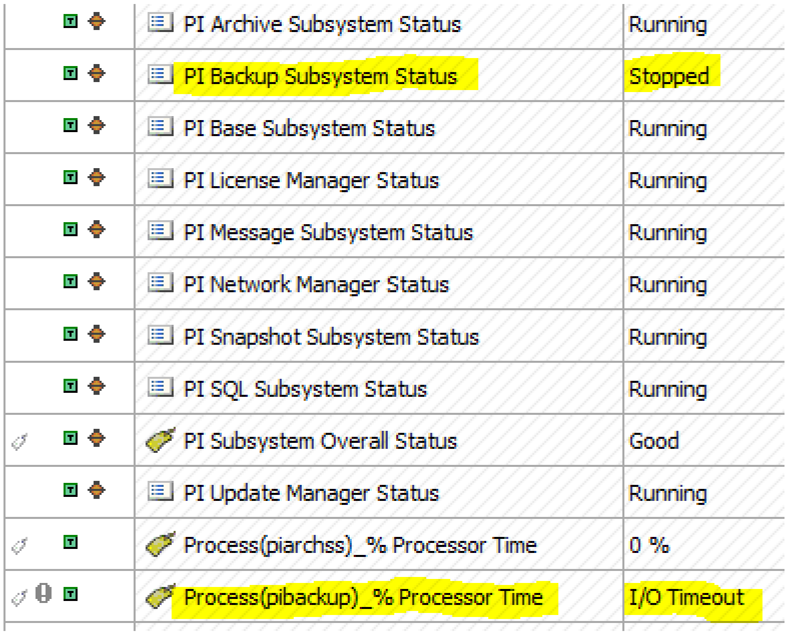
iii. Then select **Stop**. Confirm the status shows as blank (ie: not running):



PI Archive Subsystem	Running	Automatic	NT SERVICE\piarchss
PI Backup Subsystem		Automatic	NT SERVICE\pibackup
PI Base Subsystem	Running	Automatic	NT SERVICE\pibasess
PI Buffer Subsystem	Running	Automatic	PISCHOOL\svc-PIInterface
PI License Manager	Running	Automatic	NT SERVICE\pilicmgr

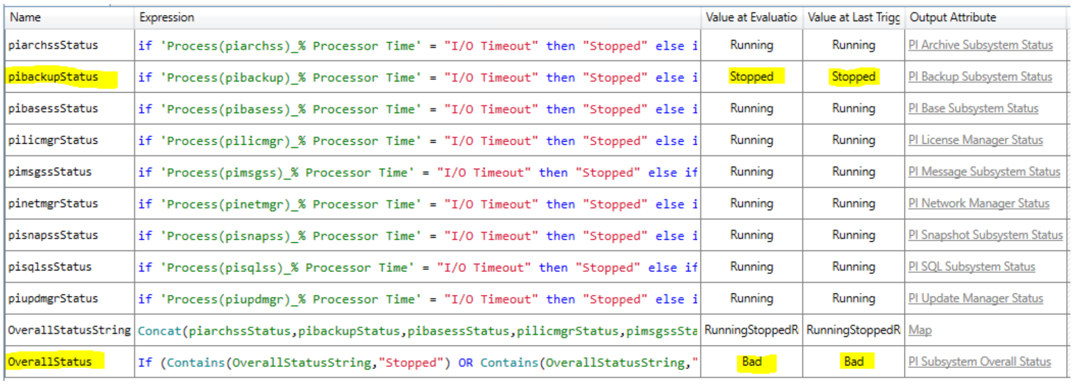
iv. Return to your web browser connection for PIMONITOR01

d. Back on PIMONITOR01, in the **Attributes** tab look at the value for the **Process(pibackup)_% Processor Time** PI point data reference attribute – it shows as **I/O Timeout** indicating the service has stopped running. Similarly the value for the **PI Backup Subsystem Status** variable has changed to **Stopped**. You may need to hit **Refresh** to see the new values. Note that the value for **PI Subsystem Overall Status** may still be **Good** – we will look at why.



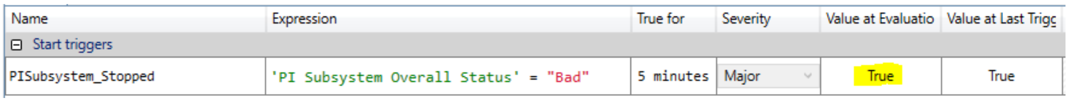
PI Archive Subsystem Status	Running
PI Backup Subsystem Status	Stopped
PI Base Subsystem Status	Running
PI License Manager Status	Running
PI Message Subsystem Status	Running
PI Network Manager Status	Running
PI Snapshot Subsystem Status	Running
PI SQL Subsystem Status	Running
PI Subsystem Overall Status	Good
PI Update Manager Status	Running
Process(piarchss)_% Processor Time	0 %
Process(pibackup)_% Processor Time	I/O Timeout

i. Select the **Analyses** tab and select the **PI Subsystem Status** expression analysis. Click **Evaluate** and see that the value at evaluation for **pibackupStatus** shows as **Stopped**, while the value at last trigger may show as **Running** or **Stopped**. This is because the analysis is scheduled periodically and runs every 1 minute. If you happened to check the evaluation before the scheduled run you may have seen it as still **Running** because it hadn't triggered yet after you stopped the subsystem. Once it triggers, the value at last trigger for **pibackupStatus** will show as **Stopped** which then sets the value for **OverallStatus** to **Bad** indicating a subsystem has failed.



Name	Expression	Value at Evaluatio	Value at Last Trigg	Output Attribute
piarchssStatus	if 'Process(piarchss)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI Archive Subsystem Status
pibackupStatus	if 'Process(pibackup)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Stopped	Stopped	PI Backup Subsystem Status
pibasessStatus	if 'Process(pibasess)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI Base Subsystem Status
pilicmgrStatus	if 'Process(pilicmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI License Manager Status
pimsgssStatus	if 'Process(pimsgss)_% Processor Time' = "I/O Timeout" then "Stopped" else if	Running	Running	PI Message Subsystem Status
pinetmgrStatus	if 'Process(pinetmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI Network Manager Status
pisnapssStatus	if 'Process(pisnapss)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI Snapshot Subsystem Status
pisqlssStatus	if 'Process(pisqlss)_% Processor Time' = "I/O Timeout" then "Stopped" else if	Running	Running	PI SQL Subsystem Status
piupdmgrStatus	if 'Process(piupdmgr)_% Processor Time' = "I/O Timeout" then "Stopped" else i	Running	Running	PI Update Manager Status
OverallStatusString	Concat(piarchssStatus,pibackupStatus,pibasessStatus,pilicmgrStatus,pimsgssSta	RunningStoppedR	RunningStoppedR	Map
OverallStatus	If (Contains(OverallStatusString,"Stopped") OR Contains(OverallStatusString,"	Bad	Bad	PI Subsystem Overall Status

ii. Select the **PI Subsystem Failure** event frame analysis. Click **Evaluate** and see that the value is **True**. Note that it has to stay true for 5 mins for the event to trigger.

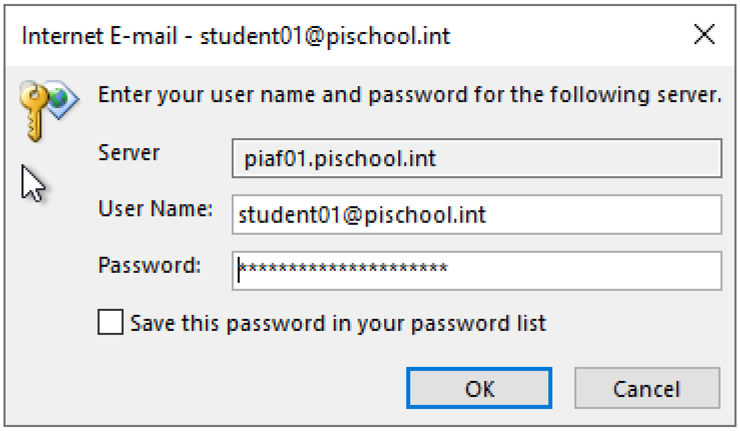


Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
⊟ Start triggers					
PISubsystem_Stopped	'PI Subsystem Overall Status' = "Bad"	5 minutes	Major	True	True

e. While you are waiting for the event frame to trigger, open Microsoft Outlook on the PIMONITOR01 machine.

i. When you open Outlook It will prompt you with a Microsoft Office Activation Wizard popup – click **Close**.

ii. It will then prompt you to log in to the student01 and student02 accounts. **Replace the default password** on the login screens with the password for your system and click **OK** (for both).



Internet E-mail - student01@pischool.int

Enter your user name and password for the following server.

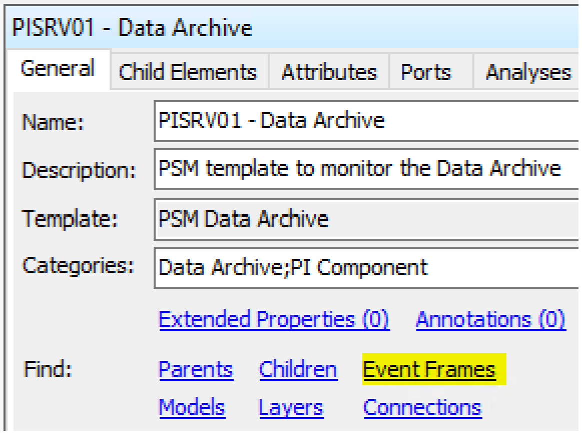
Server: piaf01.pischool.int

User Name: student01@pischool.int

Password: ********************

□ Save this password in your password list

f. By now you should have an event frame. In PI System Explorer, click on the **General** tab in the viewer, and click the **Event Frames** link to see any event frames created by this element:



PISRV01 - Data Archive

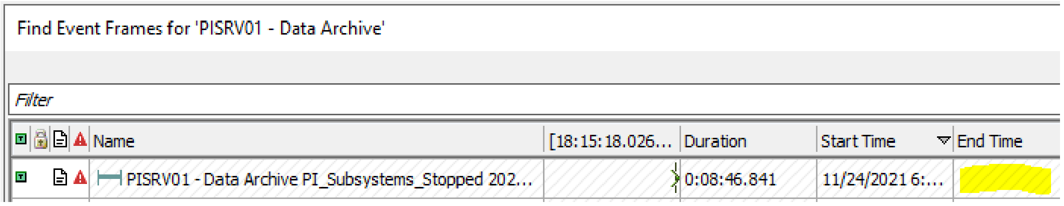
Name: PISRV01 - Data Archive

Description: PSM template to monitor the Data Archive

Template: PSM Data Archive

Categories: Data Archive;PI Component

i. Once the new event frame has been created it will show in the **Find Event Frames** window. You should see a new event frame for **Data Archive PI_Subsystems_Stopped**. You can see it is currently active (not yet closed) because the **End Time** is blank

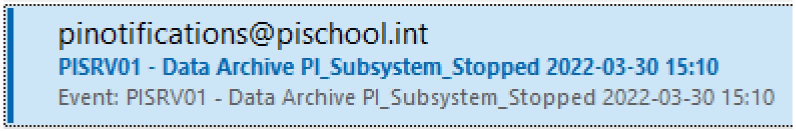


Find Event Frames for 'PISRV01 - Data Archive'

Name	[18:15:18.026…	Duration	Start Time	End Time
PISRV01 - Data Archive PI_Subsystems_Stopped 202…		0:08:46.841	11/24/2021 6:…	

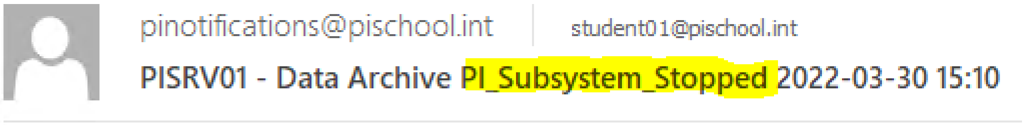
ii. **Close** the **Event Frame** window.

g. In Outlook, select the *student01@pischool.int* Inbox and locate the email notification for the **Data Archive PI_Subsystems_Stopped** notification. If it is not showing yet, click **Send/Receive All Folders** on the top right of the **Home** Toolbar to refresh.

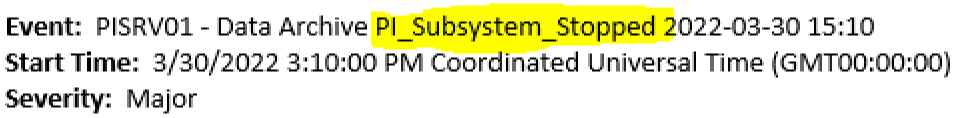


pinotifications@pischool.int
PISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10
Event: PISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10

i. Open the email and look at the content. The event name is shown in both the subject and body of the email, and includes the **Start Trigger** to show which condition triggered the event.

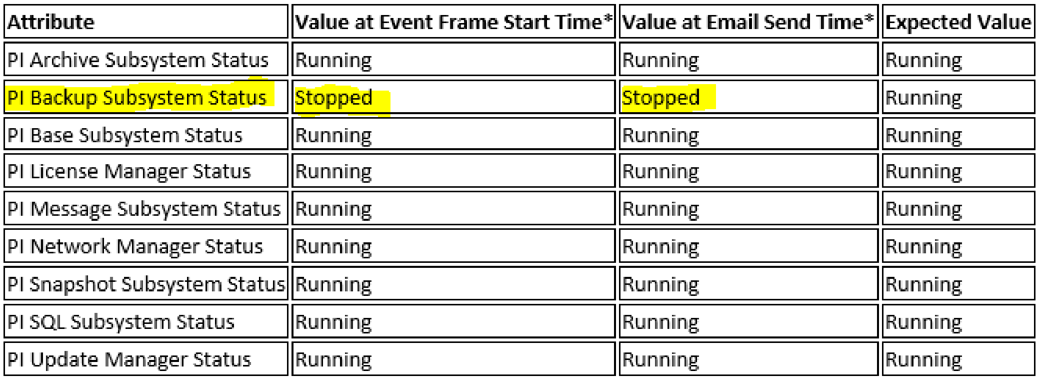


pinotifications@pischool.int | student01@pischool.int
PISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10



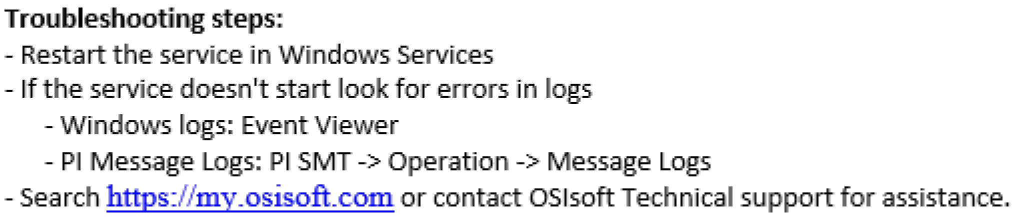
Event: PISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10
Start Time: 3/30/2022 3:10:00 PM Coordinated Universal Time (GMT00:00:00)
Severity: Major

ii. The table provides some key information on performance indicators showing the values at start and send times and expected value for evaluation. Note that is shows the PI Backup Subsystem has stopped.



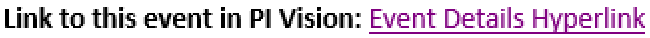
Attribute	Value at Event Frame Start Time*	Value at Email Send Time*	Expected Value
PI Archive Subsystem Status	Running	Running	Running
PI Backup Subsystem Status	Stopped	Stopped	Running
PI Base Subsystem Status	Running	Running	Running
PI License Manager Status	Running	Running	Running
PI Message Subsystem Status	Running	Running	Running
PI Network Manager Status	Running	Running	Running
PI Snapshot Subsystem Status	Running	Running	Running
PI SQL Subsystem Status	Running	Running	Running
PI Update Manager Status	Running	Running	Running

iii. Below the table are instructions to troubleshoot the issue. These can include various steps to follow to further diagnose the issue, as well as links to AVEVA documentation.



Troubleshooting steps:
- Restart the service in Windows Services
- If the service doesn't start look for errors in logs
  - Windows logs: Event Viewer
  - PI Message Logs: PI SMT -> Operation -> Message Logs
- Search https://my.osisoft.com or contact OSIsoft Technical support for assistance.

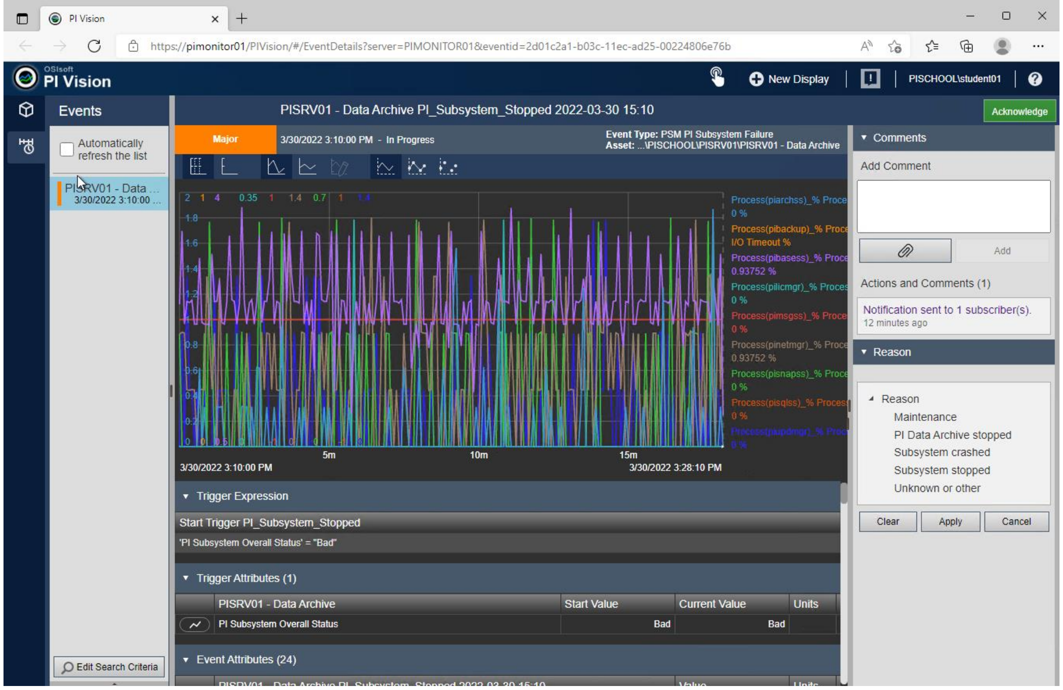
iv. Near the bottom of the email is a link to a PI Vision display for the event frame.



Link to this event in PI Vision: Event Details Hyperlink

v. Click this link in the email and Microsoft Edge will open with a PI Vision display.
NOTE: There is sometimes a bug which causes this display to not open properly the first time (it stays in a loading loop or gives a red error message). If you get to that point, click the stop and refresh buttons on the browser, and it should properly load the page

(looking something like below).



PISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10

PISRV01 - Data ... 3/30/2022 3:10:00 ...

Major 3/30/2022 3:10:00 PM - In Progress

Event Type: PSM PI Subsystem Failure
Asset: ...\PISCHOOL\PISRV01\PISRV01 - Data Archive

Process(piarchss)_% Proce 0 %
Process(pibackup)_% Proce I/O Timeout %
Process(pibasess)_% Proce 0.93752 %
Process(pilicmgr)_% Proces 0 %
Process(pimsgss)_% Proce 0 %
Process(pinetmgr)_% Proce 0.93752 %
Process(pisnapss)_% Proce 0 %
Process(pisqlss)_% Proces 0 %
Process(piupdmgr)_% Proc 0 %

Notification sent to 1 subscriber(s).
12 minutes ago

Reason
- Maintenance
- PI Data Archive stopped
- Subsystem crashed
- Subsystem stopped
- Unknown or other

Start Trigger PI_Subsystem_Stopped
'PI Subsystem Overall Status' = "Bad"

	PISRV01 - Data Archive	Start Value	Current Value	Units
	PI Subsystem Overall Status	Bad	Bad	

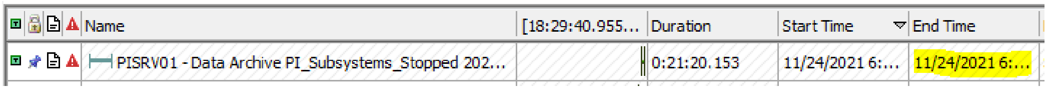
h. Return to the connection in your web browser for PISRV01

   i. On PISRV01, open the Windows **Services** pane. Find the **PI Backup Subsystem** and change the **Startup type** back to **Automatic**. Click **Apply**, then click **Start** to start the service. Click **OK** to close the window. Ensure the status changes to **Running** (you may need to click the Refresh button).

   ii. Return to the connection in your web browser for PIMONITOR01

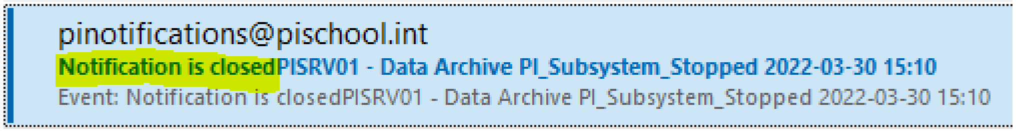
i. Back on PIMONITOR01, in PI System Explorer in the **Attributes** tab check the values of **Process(pibackup)_% Processor Time** and **PI Backup Subsystem Status** attributes – they should change back to good values (you may have to Refresh). The **PI Subsystem Overall Status** analysis output tag may still show as **Bad** – it will switch to **Good** once the scheduled analysis has run to update the value, and then the event frame will close.

   i. Select the **General** tab and click **Event Frames**. Confirm the event frame has closed – it will have an end time now.



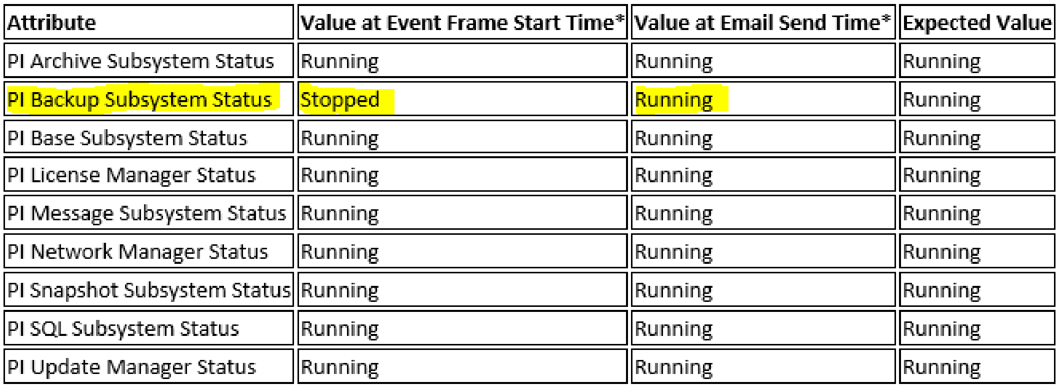
Name	[18:29:40.955...	Duration	Start Time	End Time
PISRV01 - Data Archive PI_Subsystems_Stopped 202...		0:21:20.153	11/24/2021 6:...	11/24/2021 6:...

j. You should also receive an email in Outlook indicating the event has closed. You may need to refresh to see it. The email subject is the same as the original event email with 'Notification is closed' appended to the beginning to indicate the event has ended.



pinotifications@pischool.int
Notification is closedPISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10
Event: Notification is closedPISRV01 - Data Archive PI_Subsystem_Stopped 2022-03-30 15:10

   i. Within the email body you will also see that the value at email send time has changed back to **Running**, while the value at event frame start time still shows as **Stopped** – this is so you can see which subsystem originally failed causing the notification and closure emails.



Attribute	Value at Event Frame Start Time*	Value at Email Send Time*	Expected Value
PI Archive Subsystem Status	Running	Running	Running
PI Backup Subsystem Status	Stopped	Running	Running
PI Base Subsystem Status	Running	Running	Running
PI License Manager Status	Running	Running	Running
PI Message Subsystem Status	Running	Running	Running
PI Network Manager Status	Running	Running	Running
PI Snapshot Subsystem Status	Running	Running	Running
PI SQL Subsystem Status	Running	Running	Running
PI Update Manager Status	Running	Running	Running

5. This exercise has taken you through the full end-to-end scenario of creating an element to be monitored, configuring required attributes, enabling analyses and notifications, detecting an issue, notifying you of that issue, providing troubleshooting information, and notifying you once the issue has been resolved. This illustrates a full PI System monitoring use case involving:
    a. Asset Framework to create the element (based on templates)
    b. Data Archive to hold PI Tag data
    c. PI Interface for Performance Monitor to gather data from Windows performance counters and store it in PI Tags
    d. Asset Analytics to monitor the PI Tags for incoming data, detect anomalies, and create event frames to capture the issue
    e. PI Notifications to alert on the issue
    f. PI Vision to visualize the event frame data

6. Exercise 3: PI Interface – failed interface

Exercise objectives

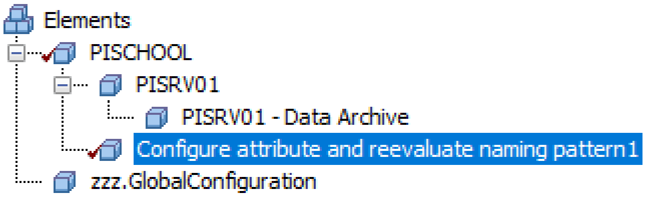
- Create and configure the elements in your AF structure for monitoring a PI Interface (you will use PI Interface for OPC in this course)
- Create required monitoring tags
- Understand the analyses used for monitoring
- Simulate issues to trigger event frames and notifications

Scenario

You will create the elements necessary for PI OPC interface monitoring (including the PIINT01 machine). After reviewing what configuration data is required for the elements to function, you will locate the required data and configure the elements appropriately. You will review the analyses and determine how they are used for monitoring, and what triggers them. You will simulate a failed interface service by triggering analyses to produce event frames and notifications, and review these outputs.

Approach

1. Refer the PI architecture diagram provided in the scenario at the beginning of this course manual to get the machine name of the interface node(s). For this exercise you will be focusing on the standalone instance of PI Interface for OPC – which machine is that installed on?
2. In this course you will only create the elements for the PIINT01 PI Interface node machine, and only for the standalone PI OPC Interface instance on it. In PI System Explorer, in the browser right-click the **PISCHOOL** element in your database and select **New > New Child Element**
    a. In the Choose Element Template window, ensure reference type **Domain-Server** is chosen, and select the **PSM Server (Basic)** template. You will see a new **Server** element under **PISCHOOL** in the element structure. For now we just need a Server element to exist so we can create the **Interface** component under it.
        i. The **PSM Server (Full)** template includes several more attributes for detailed server monitoring. If you wish to learn more about the **PSM Server (*)** templates see *Optional exercise 1: Server – high processor utilization*
    b. The element will have a default name indicating that user input is required:



Elements
PISCHOOL
PISRV01
PISRV01 - Data Archive
Configure attribute and reevaluate naming pattern1
zzz.GlobalConfiguration

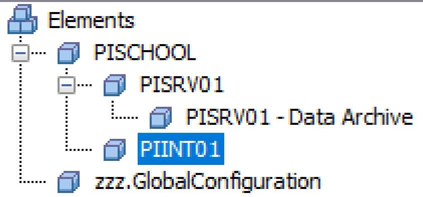
Select the **Attributes** tab in the viewer. This element uses the same template as the PISRV01 element you created previously, so the attributes and analyses will be the same. You will need to fill in certain details that were provided at the beginning of this document in the **Target** attribute:

i. Target (machine name for this server): **PIINT01**
ii. The default values for Location1 and Location 4 can stay the same

iii. PerfMon Point Source (point source for the PerfMon instance that is retrieving the performance counters for this server): **PIMONITOR01_PERF01**

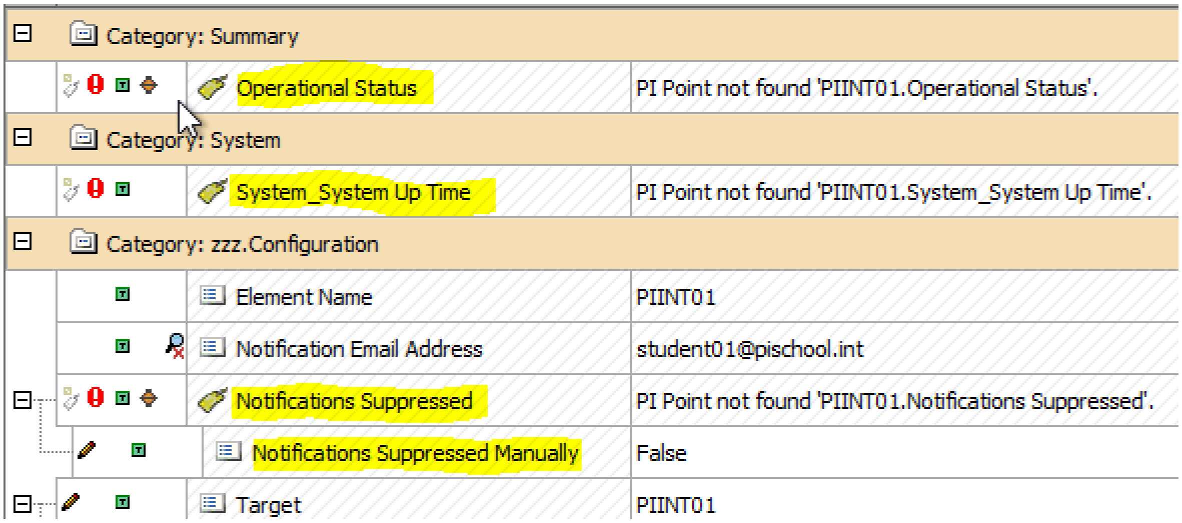
! IMPORTANT	Ensure the values you have entered look **EXACTLY** as listed above (no extra spaces).

c. Check in your changes and refresh the view. At this point your hierarchy should appear as follows



- Elements
  - PISCHOOL
    - PISRV01
      - PISRV01 - Data Archive
    - PIINT01
  - zzz.GlobalConfiguration

d. There are additional attributes on the Server template used to monitor the overall availability status of the server, and in certain cases can be used to suppress notifications when a server or component is unavailable.

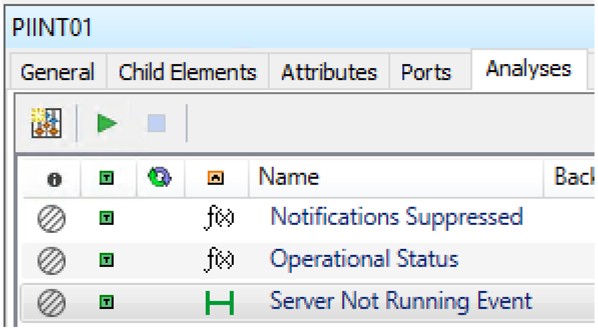


Category: Summary	
Operational Status	PI Point not found 'PIINT01.Operational Status'.
Category: System	
System_System Up Time	PI Point not found 'PIINT01.System_System Up Time'.
Category: zzz.Configuration	
Element Name	PIINT01
Notification Email Address	student01@pischool.int
Notifications Suppressed	PI Point not found 'PIINT01.Notifications Suppressed'.
Notifications Suppressed Manually	False
Target	PIINT01

To create tags for the PI Point data reference attributes:

i. Right-click on the **PIINT01** element in the browser and select **Create or Update Data Reference**. The Create Config window will show the progress of the creation.

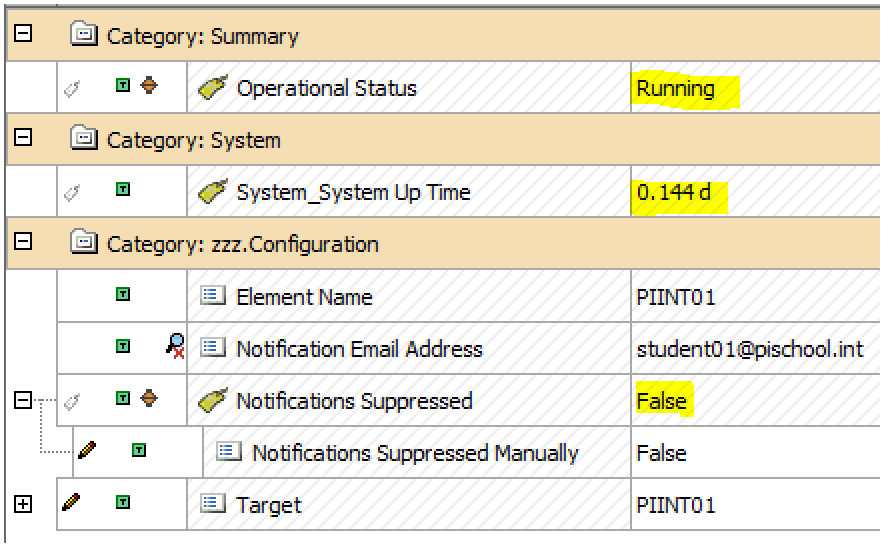
ii. Once the tags have finished creating, select the **Analyses** tab in the viewer and you will see a number of analyses defined for this element. As previously, the analyses are disabled.



Name	Bacl
Notifications Suppressed	
Operational Status	
Server Not Running Event	

iii. You can enable them individually from here or use **Management** from the navigator to enable them all at once, and confirm they have the correct status after startup. Enable the new notification rule as well.

e. Once the analyses and notification are running, go back to the **Attributes** tab, wait a few minutes, and click Refresh on the toolbar to see values. You may have to wait up to 5 minutes to see a value for **Notifications Suppressed** as that analysis is on a 5 minute periodic schedule.



Category: Summary	
Operational Status	Running
Category: System	
System_System Up Time	0.144 d
Category: zzz.Configuration	
Element Name	PIINT01
Notification Email Address	student01@pischool.int
Notifications Suppressed	False
Notifications Suppressed Manually	False
Target	PIINT01

f. This has configured and enabled overall availability status of the server. We will not be going further into this functionality at this time. See the following for more information on:

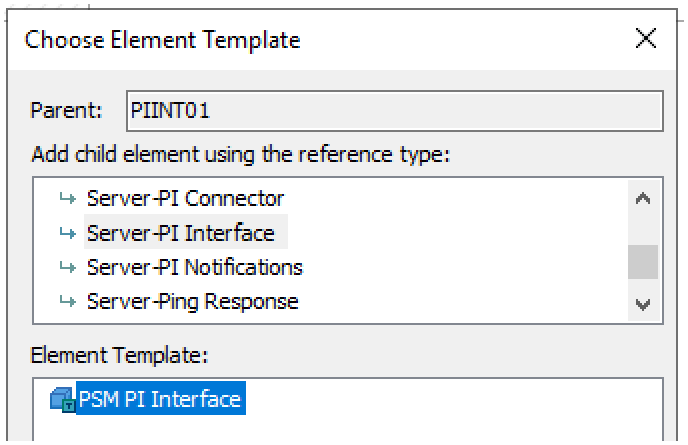
i. server-level monitoring: *Optional exercise 1: Server – high processor utilization*

ii. *Notification suppression*

3. Next you will create the element for the PI Interface component. Key things to monitor for a PI Interface include overall health of the Interface (is it up and running), low I/O rates (is it sending data), and missed/skipped scans (missing some data). Monitoring the health of your PI Interface allows you to proactively address issues that arise.

a. In PI System Explorer, in the browser right-click the **PIINT01** element in your database and select **New > New Child Element**. Note that this is an element for the interface instance, so if you had multiple interface instances (same or different types) running on the same machine, you would create an element for each.

i. In the Choose Element Template window, ensure reference type **Server-PI Interface** is chosen, and select the **PSM PI Interface** template:



Choose Element Template

Parent: PIINT01

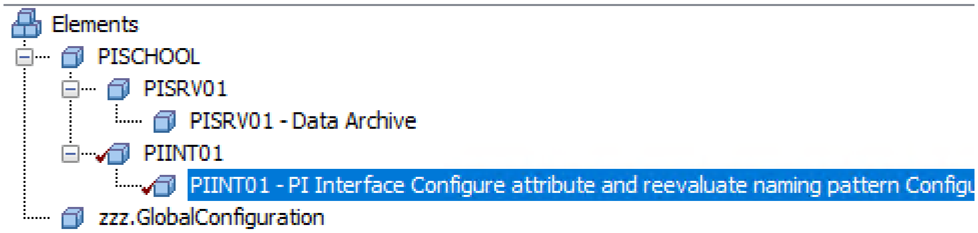
Add child element using the reference type:

- Server-PI Connector
- Server-PI Interface
- Server-PI Notifications
- Server-Ping Response

Element Template:

PSM PI Interface

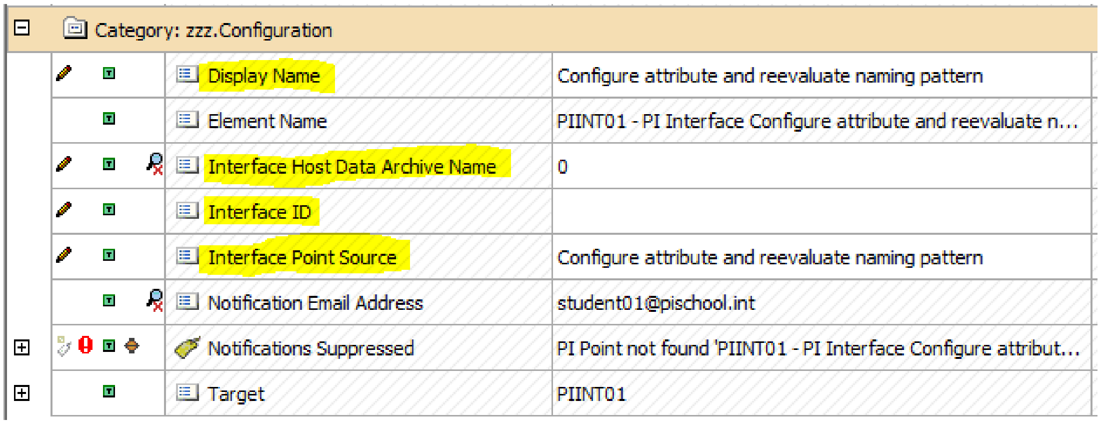
ii. You will see a new PI Interface element under PIINT01 in the element structure. The element will have a default name indicating that user input is required:



```
Elements
  PISCHOOL
    PISRV01
      PISRV01 - Data Archive
    PIINT01
      PIINT01 - PI Interface Configure attribute and reevaluate naming pattern Configu
  zzz.GlobalConfiguration
```

b. Select the **Attributes** tab. Find and expand the **Target** attribute. If you scroll down it should be at the bottom under **Category: zzz.Configuration**. Notice that the **Target** and **PerfMon Point Source** attributes are already filled in – they are inherited from the parent element.

c. You will notice there are several items under the zzz.Configuration category that require user input (✎ pencil icon to the left). In order to set the element name, you require the **Display Name** and the **Interface Point Source** of the interface. **Interface Host Data Archive Name** (which Data Archive this interface is writing data to) and **Interface ID** are also required as part of the monitoring tag configuration:



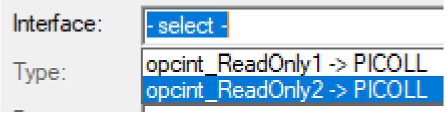
Category: zzz.Configuration

✎	Display Name	Configure attribute and reevaluate naming pattern
	Element Name	PIINT01 - PI Interface Configure attribute and reevaluate n...
✎	Interface Host Data Archive Name	0
✎	Interface ID	
✎	Interface Point Source	Configure attribute and reevaluate naming pattern
	Notification Email Address	student01@pischool.int
	Notifications Suppressed	PI Point not found 'PIINT01 - PI Interface Configure attribut...
	Target	PIINT01

i. You must retrieve the required configuration data from the interface itself which is installed on the PIINT01 machine.

1. In your web browser, connect to the PIINT01 server.

2. On PIINT01, click on the ICU icon on the bottom taskbar to open ICU (Interface Configuration Utility).
3. Select **opcint_ReadOnly2 -> PISRV01**

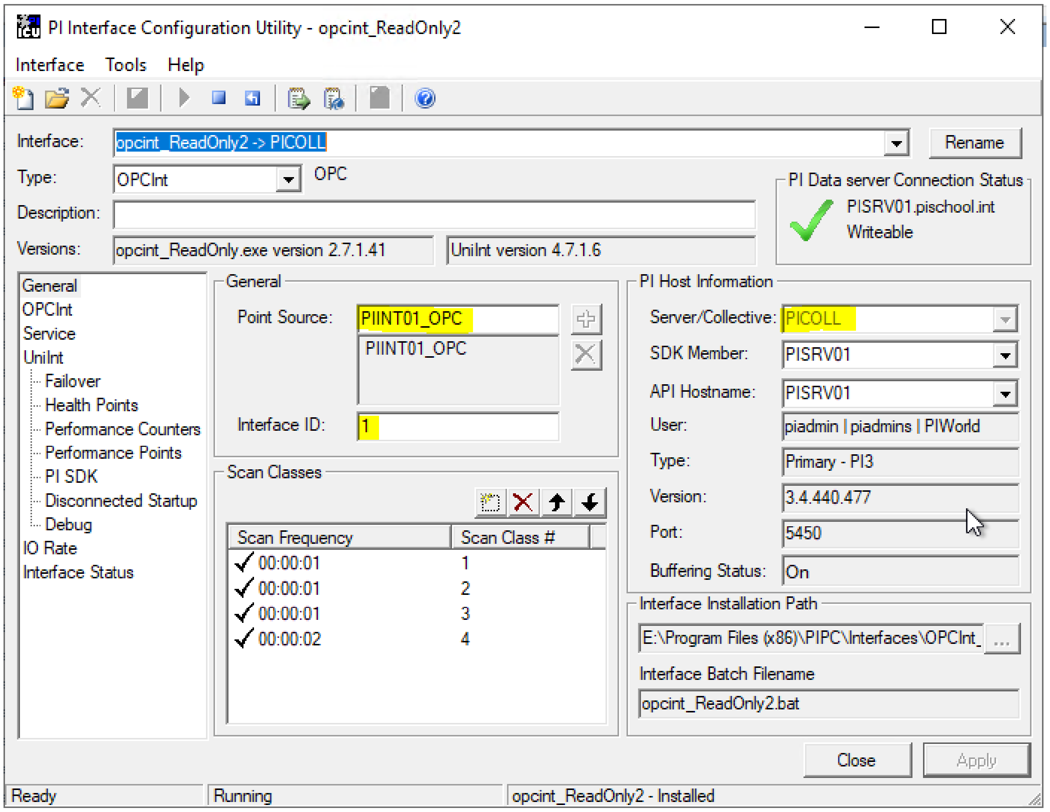


Interface: - select -

Type:

opcint_ReadOnly1 -> PICOLL
opcint_ReadOnly2 -> PICOLL

4. On the **General** tab, note the
   a. Point Source: **PIINT01_OPC**
   b. Interface ID: **1**
   c. Server/Collective: **PICOLL**



Interface: opcint_ReadOnly2 -> PICOLL

Type: OPCInt OPC

Description:

Versions: opcint_ReadOnly.exe version 2.7.1.41 UniInt version 4.7.1.6

PI Data server Connection Status: PISRV01.pischool.int Writeable

General

Point Source: PIINT01_OPC

PIINT01_OPC

Interface ID: 1

Scan Classes

Scan Frequency	Scan Class #
✓ 00:00:01	1
✓ 00:00:01	2
✓ 00:00:01	3
✓ 00:00:02	4

PI Host Information

Server/Collective: PICOLL

SDK Member: PISRV01

API Hostname: PISRV01

User: piadmin | piadmins | PIWorld

Type: Primary - PI3

Version: 3.4.440.477

Port: 5450

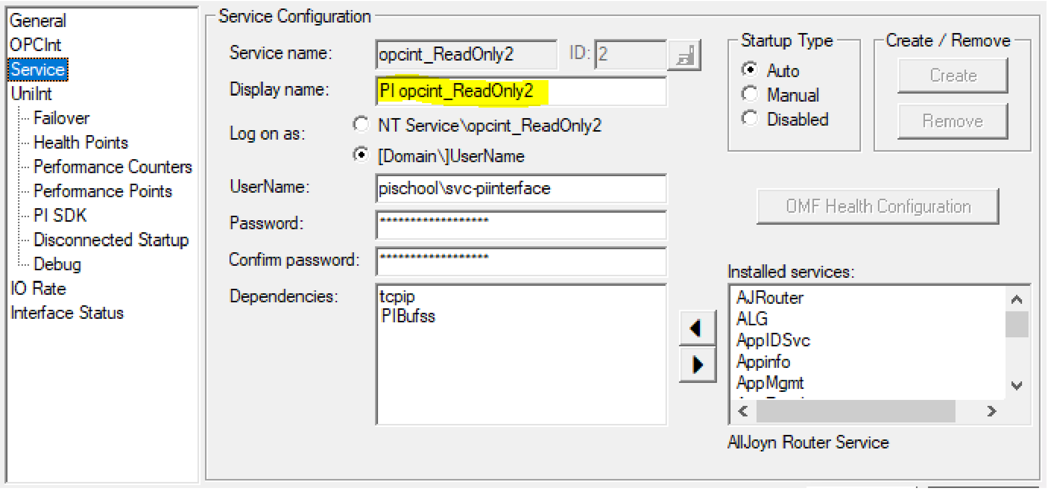
Buffering Status: On

Interface Installation Path: E:\Program Files (x86)\PIPC\Interfaces\OPCInt_

Interface Batch Filename: opcint_ReadOnly2.bat

Ready | Running | opcint_ReadOnly2 - Installed

5. On the **Service** tab, note the:
   a. Display name: **PI opcint_ReadOnly2**
   (*NOTE: there is a space after 'PI')



Service Configuration

Service name: opcint_ReadOnly2 ID: 2

Display name: PI opcint_ReadOnly2

Log on as: ○ NT Service\opcint_ReadOnly2 ● [Domain\]UserName

UserName: pischool\svc-piinterface

Password: ******************

Confirm password: ******************

Dependencies: tcpip PIBufss

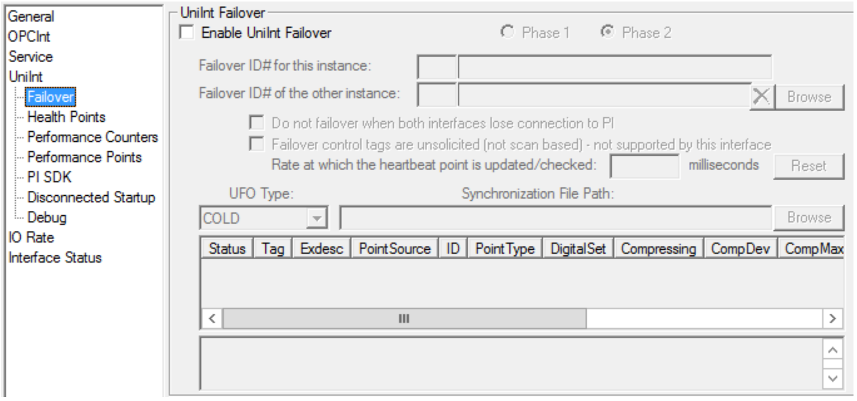
Startup Type: ● Auto ○ Manual ○ Disabled

Create / Remove

Installed services: AJRouter ALG AppIDSvc Appinfo AppMgmt

AllJoyn Router Service

6. On the Failover tab, note that failover is not enabled:



UniInt Failover

☐ Enable UniInt Failover ○ Phase 1 ● Phase 2

Failover ID# for this instance:

Failover ID# of the other instance:

☐ Do not failover when both interfaces lose connection to PI

☐ Failover control tags are unsolicited (not scan based) - not supported by this interface

Rate at which the heartbeat point is updated/checked: milliseconds

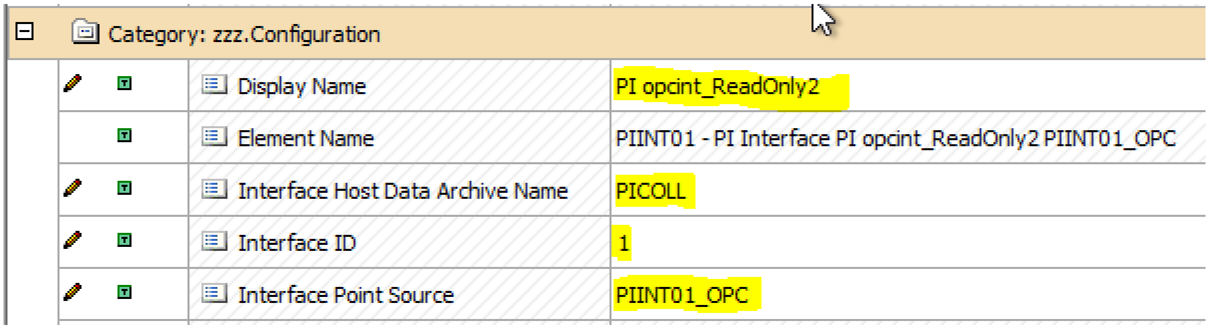
UFO Type: COLD Synchronization File Path:

Status	Tag	Exdesc	PointSource	ID	PointType	DigitalSet	Compressing	CompDev	CompMax

ii. Return to your connection for PIMONITOR01

d. Select the **Attributes** tab in the viewer. Fill in the values for the interface configuration items you retrieved from ICU. It's very important to get these correct or the tags won't be created properly, and they will not link to the Windows performance counters to get the monitoring data:

i. Note that you do not fill anything in for Element Name – it will automatically update once you enter the other values and check in your changes.

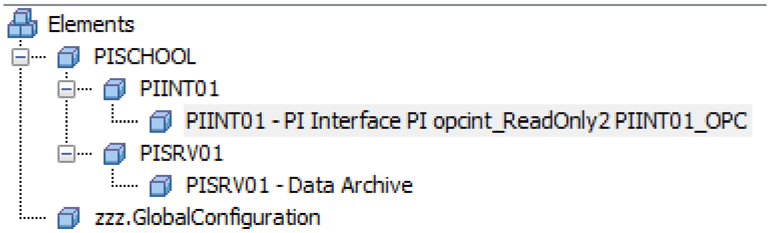


Category: zzz.Configuration

Display Name	PI opcint_ReadOnly2
Element Name	PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
Interface Host Data Archive Name	PICOLL
Interface ID	1
Interface Point Source	PIINT01_OPC

! IMPORTANT	Ensure the values you have entered look **EXACTLY** as pictured above (including the space after 'PI' in the **Display Name**).

i. Check in your changes then select Refresh from the Toolbar. At this point your hierarchy should appear as follows

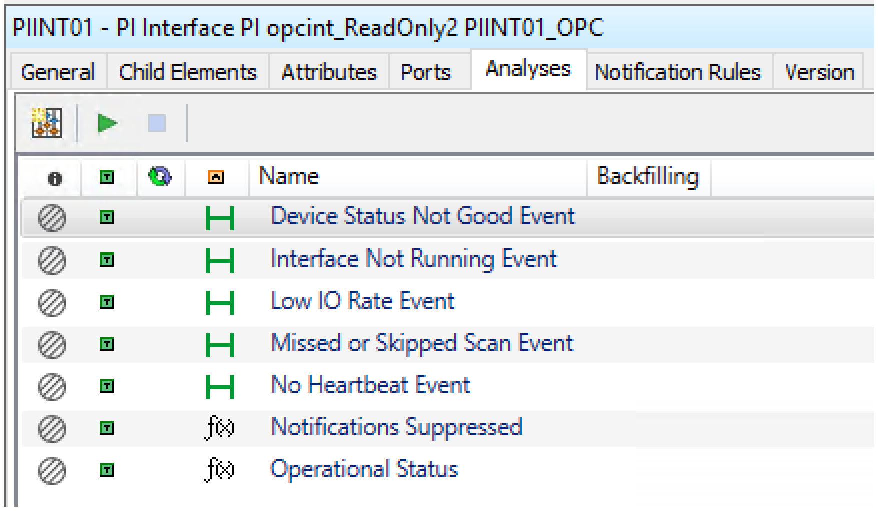

Elements
PISCHOOL
PIINT01
PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
PISRV01
PISRV01 - Data Archive
zzz.GlobalConfiguration

e. Similar to the Data Archive element, there are several attributes which are PI Point data references. You will need to create the tags for these.

i. Right-click on the **PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC** element in the browser and select **Create or Update Data Reference**. The Create Config window will show the progress of the creation.

ii. Once the tags have finished creating, wait a few minutes, and click Refresh on the toolbar to see values.

iii. Select the **Analyses** tab in the viewer and you will see several analyses defined for this element. As previously, the analyses are disabled.

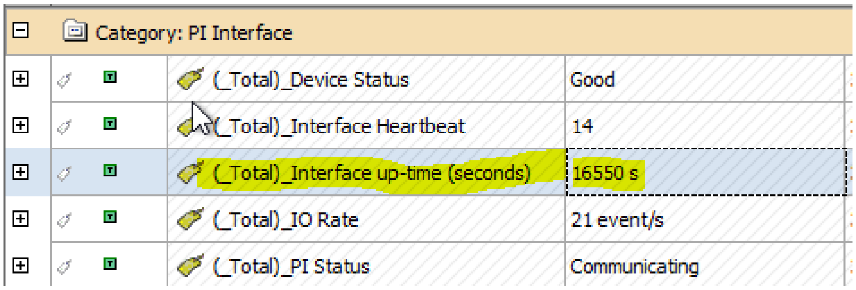

PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC

Name	Backfilling
Device Status Not Good Event	
Interface Not Running Event	
Low IO Rate Event	
Missed or Skipped Scan Event	
No Heartbeat Event	
Notifications Suppressed	
Operational Status	

iv. You can enable them individually from here or use **Management** from the navigator to enable them all at once, and confirm they have the correct status after startup. Enable any new notification rules as well.

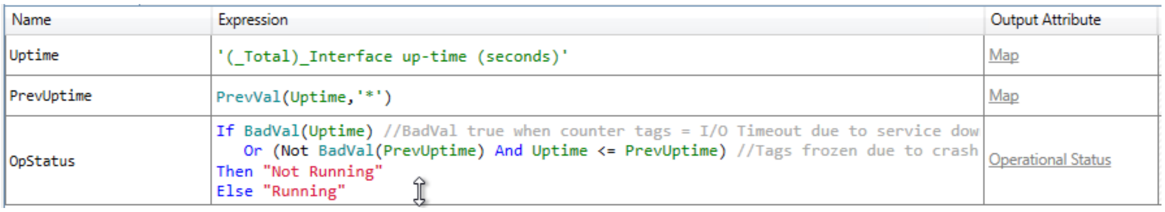
4. Now that you have created and configured the elements and analyses, you will simulate a failed or stopped interface service. An interface service may stop for various reasons - server reboot, service dependency, person forgot to restart after scheduled maintenance, failure, etc. It is important to know when this has happened so it can be addressed quickly to prevent potential data loss or data unavailability.

a. Select the **Attributes** tab in the viewer. Take a look at the **(_Total)_Interface up-time (seconds)** attribute – this is a PI Point data reference linked to a Windows performance counter that logs the total time the interface has been running since it was last started:



Category: PI Interface	
(_Total)_Device Status	Good
(_Total)_Interface Heartbeat	14
(_Total)_Interface up-time (seconds)	16550 s
(_Total)_IO Rate	21 event/s
(_Total)_PI Status	Communicating

b. Go to the **Analyses** tab in the viewer and select the **Operational Status** expression. In the lower window you will see three variables defined:



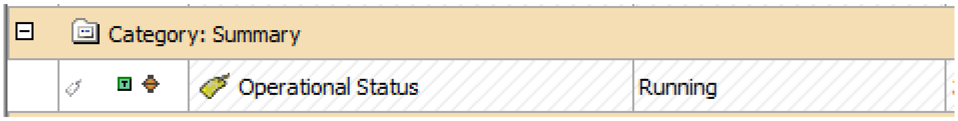
Name	Expression	Output Attribute
Uptime	'(_Total)_Interface up-time (seconds)'	Map
PrevUptime	PrevVal(Uptime,'*')	Map

OpStatus	If BadVal(Uptime) //BadVal true when counter tags = I/O Timeout due to service dow
	Or (Not BadVal(PrevUptime) And Uptime <= PrevUptime) //Tags frozen due to crash
	Then "Not Running"
	Else "Running"
		Operational Status

i. Variable **Uptime** is getting the value for the **(_Total)_Interface up-time (seconds)** attribute we just looked at

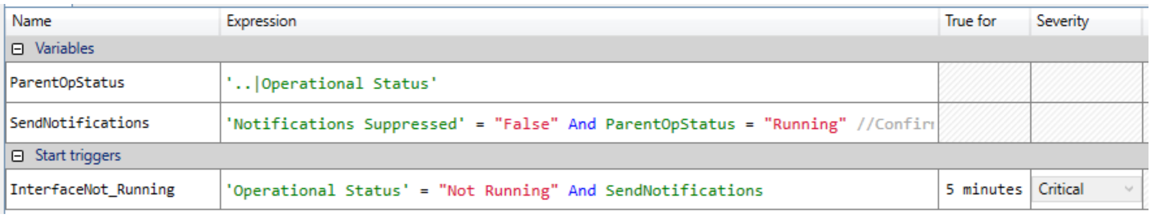
ii. Variable **PrevUptime** is getting the previous PI Tag value for the same attribute as above

iii. Variable **OpStatus** uses the above two variables to see if the service has stopped or crashed, and sets the **Operational Status** analysis output attribute value accordingly, which you can see back on the **Attributes** tab:



Category: Summary	
Operational Status	Running

c. On the **Analyses** tab select the **Interface Not Running Event** event frame. In the lower window you will see two variables and a start trigger defined:



Name	Expression	True for	Severity
Variables			
ParentOpStatus	'..|Operational Status'		
SendNotifications	'Notifications Suppressed' = "False" And ParentOpStatus = "Running" //Confir		
Start triggers			
InterfaceNot_Running	'Operational Status' = "Not Running" And SendNotifications	5 minutes	Critical

i. Variable **ParentOpStatus** is getting the value for the **Operational Status** attribute of the parent element (which we reviewed in the previous step) - is the server running?

ii. Variable **SendNotifications** is checking various conditions to see if it should be sending notifications:

1. Have notifications been suppressed - the interface's own **Notifications Suppressed** attribute would be **True** if it had been suppressed manually or if its parent (the server) were suppressed, and **False** otherwise. See *Notification suppression* for more information.
2. The value of **ParentOpStatus** from above

iii. The start trigger **Interface_Not_Running** is checking if:

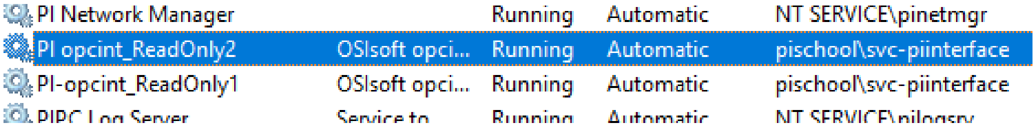
1. The interface's own **Operational Status** attribute value indicates a problem (stopped or crashed)
2. **SendNotifications** variable from above indicates that notifications should be sent

If both of these are true for at least 5 minutes, then the event frame is triggered.

iv. The analysis is event-triggered and evaluates every time a new value is received for any of the inputs.

d. Return to your connection for PIINT01

i. On PIINT01, open the Windows **Services** pane via the icon on the taskbar. Find the service that matches the **Display Name** for the interface instance – it should be **Running**.



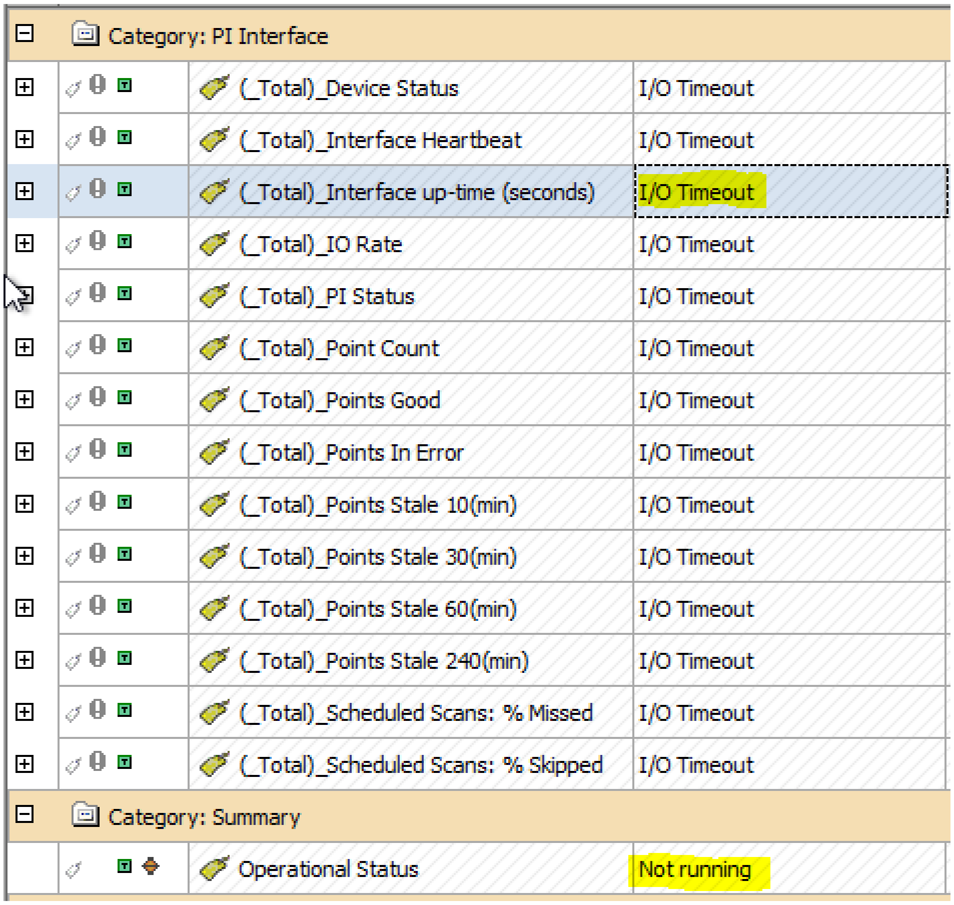
PI Network Manager		Running	Automatic	NT SERVICE\pinetmgr
PI opcint_ReadOnly2	OSIsoft opci...	Running	Automatic	pischool\svc-piinterface
PI-opcint_ReadOnly1	OSIsoft opci...	Running	Automatic	pischool\svc-piinterface
PIPC Log Server	Service to	Running	Automatic	NT SERVICE\pilogsrv

ii. Right-click on the service and select **Properties**. Set Startup type to **Disabled** and click **Apply**. (We have a script running on the course machines which catches any PI-related service failures and restarts them automatically, so we need to disable the service so it can't restart.)

iii. Then select **Stop**. Confirm the status shows as blank (ie: not running).

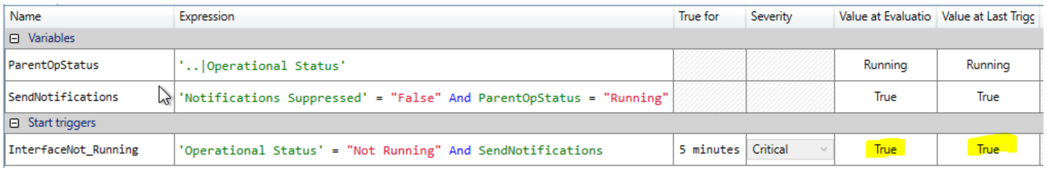
iv. Return to your connection for PIMONITOR01

e. Back on PIMONITOR01 in PI System Explorer, click Refresh and confirm the value for the **(_Total)_Interface up-time (seconds)** attributes should show as **I/O Timeout** indicating no values are incoming for that tag. Similarly all other monitoring tags will also show **I/O Timeout** when the interface is fully stopped. **Operational Status** should show as **Not running**



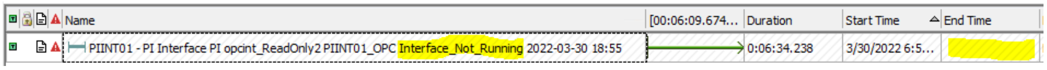
Category: PI Interface	
(_Total)_Device Status	I/O Timeout
(_Total)_Interface Heartbeat	I/O Timeout
(_Total)_Interface up-time (seconds)	I/O Timeout
(_Total)_IO Rate	I/O Timeout
(_Total)_PI Status	I/O Timeout
(_Total)_Point Count	I/O Timeout
(_Total)_Points Good	I/O Timeout
(_Total)_Points In Error	I/O Timeout
(_Total)_Points Stale 10(min)	I/O Timeout
(_Total)_Points Stale 30(min)	I/O Timeout
(_Total)_Points Stale 60(min)	I/O Timeout
(_Total)_Points Stale 240(min)	I/O Timeout
(_Total)_Scheduled Scans: % Missed	I/O Timeout
(_Total)_Scheduled Scans: % Skipped	I/O Timeout
Category: Summary	
Operational Status	Not running

i. Select the **Analyses** tab and select the **Interface Not Running Event** event frame analysis. Click **Evaluate** and see that the evaluation for the start trigger is **True**, indicating there is an interface issue.



Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
⊟ Variables					
ParentOpStatus	'..|Operational Status'			Running	Running
SendNotifications	'Notifications Suppressed' = "False" And ParentOpStatus = "Running"			True	True
⊟ Start triggers					
InterfaceNot_Running	'Operational Status' = "Not Running" And SendNotifications	5 minutes	Critical	True	True

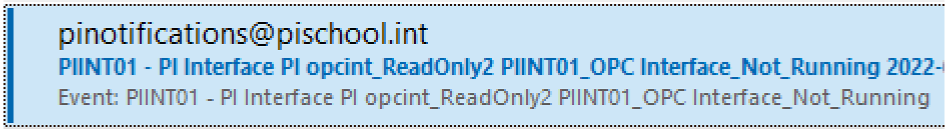
ii. Select the **General** tab and click **Event Frames** – after the event trigger condition has been true for 5 minutes, you will see a new event frame for **Interface Not Running Event**. The blank **End Time** shows the event frame is still active.



Name	[00:06:09.674...	Duration	Start Time	End Time
PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Interface_Not_Running 2022-03-30 18:55		0:06:34.238	3/30/2022 6:5...	

iii. **Close** the **Event Frame** window.

f. Open Microsoft Outlook on the PIMONITOR01 machine. In the *student01@pischool.int* Inbox locate the email for the **Interface Not Running Event** notification. Refresh if it is not showing.



pinotifications@pischool.int
PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Interface_Not_Running 2022-
Event: PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Interface_Not_Running

i. Open the email and look at the content. The event name (which is also the email subject) indicates the name of the triggering condition (**Interface_Not_Running**). The table provides some key information on performance indicators. Below the table are instructions to troubleshoot the issue and a link to the event details view in PI Vision.

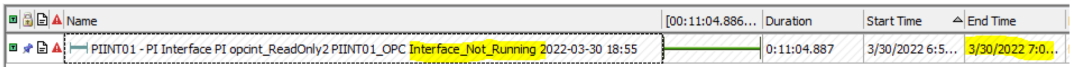
g. Return to PIINT01

i. On PIINT01, open the Windows **Services** pane. Find the service that matches the **Display Name** for the interface instance and change the **Startup type** back to **Automatic**. Click **Apply**, then click **Start** to start the service. Click **OK** to close the window. Ensure the status changes to **Running** (you may need to click the Refresh button).

ii. Return to PIMONITOR01

h. Back on PIMONITOR01, in PI System Explorer in the **Attributes** tab check the values of **(_Total)_Interface up-time (seconds)** and **Operational Status** attributes – they should change back to good values (you may have to refresh and wait a few minutes for the interface to connect and start sending data).

i. Select the **General** tab and click **Event Frames**. Confirm the event frame has closed – it will have an **End Time** now.



Name	[00:11:04.886...	Duration	Start Time	End Time
PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Interface_Not_Running 2022-03-30 18:55		0:11:04.887	3/30/2022 6:5...	3/30/2022 7:0...

ii. You should also receive an email in Outlook indicating the event has closed. You may need to refresh to see it.

7. Exercise 4: Analysis service – high latency

Exercise objectives

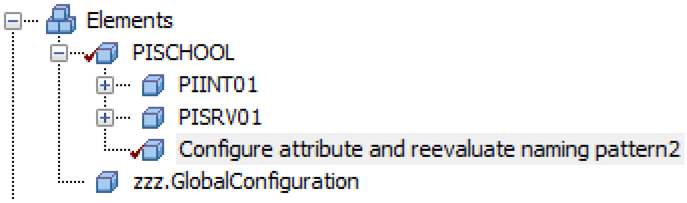
- Create and configure the elements in your AF structure for monitoring Asset Analytics
- Create required monitoring tags
- Understand the analyses used for monitoring
- Simulate a high latency condition to trigger event frames and notifications

Scenario

You will create the elements necessary for Asset Analytics monitoring (including the PIAF01 machine). After reviewing what configuration data is required for the elements to function, you will locate the required data and configure the elements appropriately. You will review the analyses and determine how they are used for monitoring, and what triggers them. You will simulate a high latency condition (delays in calculations) by triggering analyses to produce event frames and notifications, and review these outputs.

Approach

1. Review the PI architecture diagram provided in the Scenario at the beginning of this course manual.
   a. Where is Asset Analytics installed?
2. First you will create the PIAF01 machine element. For now we just need a Server element to exist so we can create the **Asset Analytics** component under it, so we will use the **PSM Server (Basic)** template. In PI System Explorer, in the browser right-click the **PISCHOOL** element in your database and select **New > New Child Element**
   a. In the Choose Element Template window, ensure reference type **Domain-Server** is chosen, and select the **PSM Server (Basic)** template. You will see a new **Server** element under **PISCHOOL** in the element structure. The element will have a default name indicating that user input is required:



   b. Select the **Attributes** tab in the viewer. This element uses the same template as the PISRV01 and PIINT01 elements you created previously, so the attributes and analyses will be the same. You will need to fill in certain details that were provided at the beginning of this document in the **Target** attribute:
      i. Target (machine name for this server): **PIAF01**
      ii. The default values for Location1 and Location 4 can stay the same
      iii. PerfMon Point Source (point source for the PerfMon instance that is retrieving the performance counters for this server): **PIMONITOR01_PERF01**
      iv. Check in your changes

! IMPORTANT	Ensure the values you have entered look **EXACTLY** as listed above (no extra spaces).

c. There are additional attributes on the Server template used to monitor the overall availability status of the server, and in certain cases can be used to suppress notifications when a server or component is unavailable. To create tags for the PI Point data reference attributes:

   i. Right-click on the **PIAF01** element in the browser and select **Create or Update Data Reference**. The Create Config window will show the progress of the creation.

   ii. Once the tags have finished creating, select the **Analyses** tab in the viewer and you will see a number of analyses defined for this element. As previously, the analyses are disabled.

   iii. You can enable them individually from here or use **Management** from the navigator to enable them all at once, and confirm they have the correct status after startup. Enable the new notification rule as well.

   iv. Once the analyses and notification are running, go back to the **Attributes** tab, wait a few minutes, and click Refresh on the toolbar to see values. You may have to wait up to 5 minutes to see a value for **Notifications Suppressed** as that analysis is on a 5 minute periodic schedule.

d. This has configured and enabled overall availability status of the server. We will not be going further into this functionality at this time. See the following for more information on:

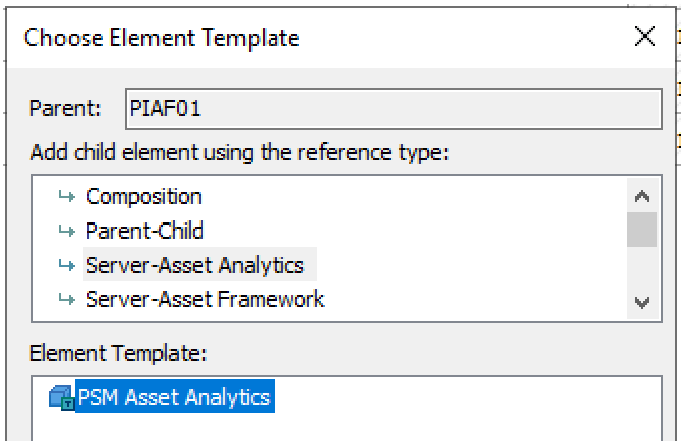
   i. server-level monitoring: *Optional exercise 1: Server – high processor utilization*

   ii. *Notification suppression*

3. Next you will create the element for the Asset Analytics component. Asset Analytics is the analytics engine for the PI System. It executes analyses, creates event frames, and backfills/recalculates data. An overloaded Asset Analytics will result in lagging calculations, skipped evaluations, and slow backfilling/recalculations. Monitoring the health of Asset Analytics allows you to proactively address issues that may arise.

   a. In PI System Explorer, in the browser right-click the **PIAF01** element in your database and select **New Child Element**.

      i. In the Choose Element Template window, ensure reference type **Server-Asset Analytics** is chosen, and select the **PSM Asset Analytics** template:



Choose Element Template

Parent: PIAF01

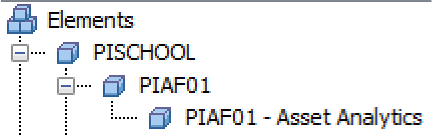
Add child element using the reference type:

- Composition
- Parent-Child
- Server-Asset Analytics
- Server-Asset Framework

Element Template:

PSM Asset Analytics

ii. You will see a new PI Analytics element with a default name under PIAF01 in the element structure. This name can stay as-is:



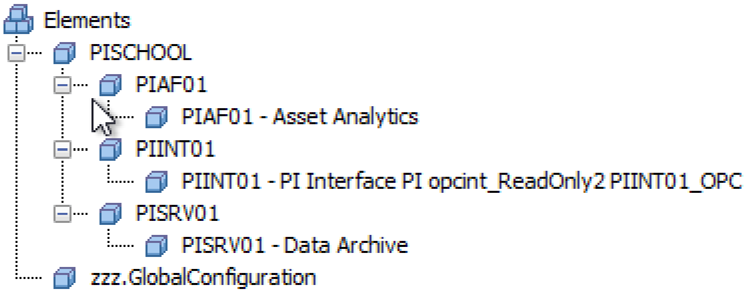
```
Elements
  PISCHOOL
    PIAF01
      PIAF01 - Asset Analytics
```

iii. Select the **Attributes** tab. Find and expand the **Target** attribute. If you scroll down it should be at the bottom under **Category: zzz.Configuration**. Notice that the **Target** and **PerfMon Point Source** attributes are already filled in – they are inherited from the parent element.

iv. Check the **zzz.Configuration** category to see if any other configuration items need to be populated. (There are none in this case.)

b. Check in your changes, then select Refresh from the toolbar.

i. Your element hierarchy should now look like this:



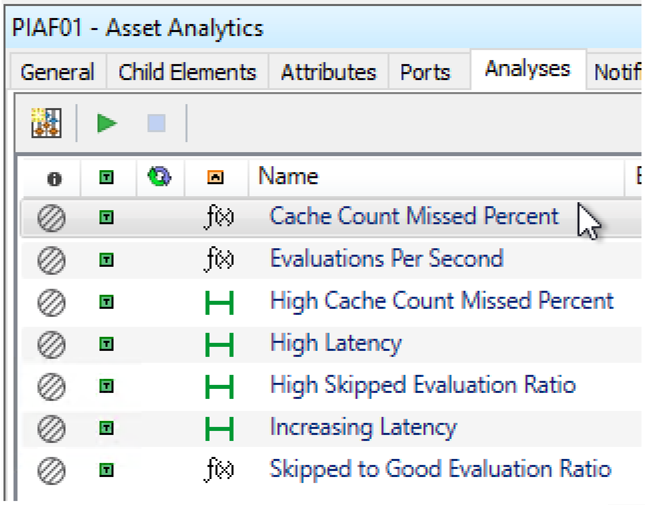
```
Elements
  PISCHOOL
    PIAF01
      PIAF01 - Asset Analytics
    PIINT01
      PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
    PISRV01
      PISRV01 - Data Archive
    zzz.GlobalConfiguration
```

c. This element has several PI Point data reference attributes. You will need to create the tags for these.

i. Right-click on the **PIAF01 – Asset Analytics** element in the browser and select **Create or Update Data Reference**. The Create Config window will show the progress of the creation.

ii. Once the tags have finished creating, wait a few minutes, and click Refresh on the toolbar to see values.

d. Select the **Analyses** tab in the viewer and you will see a number of analyses defined for this element. As previously, the analyses are disabled.



PIAF01 - Asset Analytics

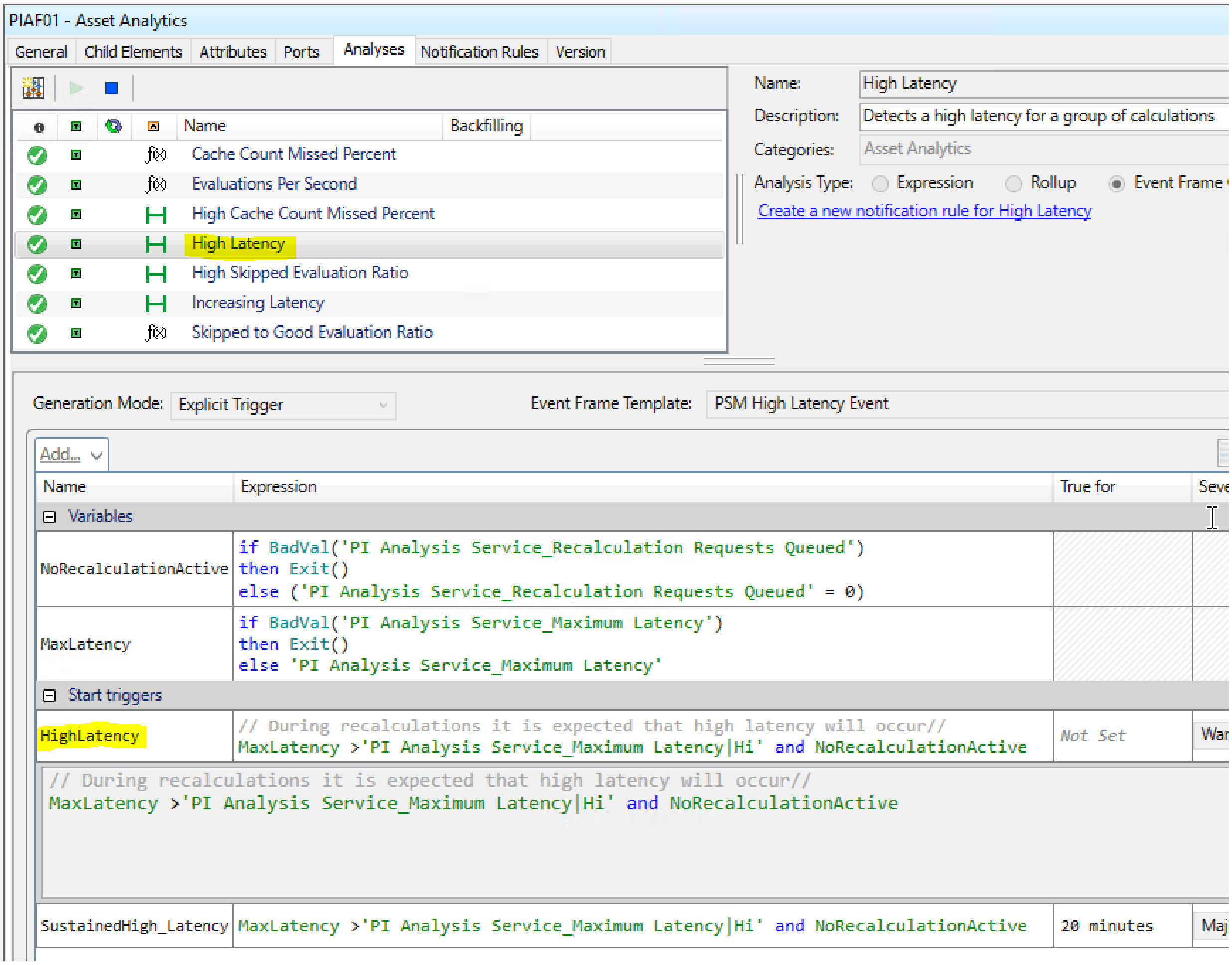
Name
Cache Count Missed Percent
Evaluations Per Second
High Cache Count Missed Percent
High Latency
High Skipped Evaluation Ratio
Increasing Latency
Skipped to Good Evaluation Ratio

e. You can enable them individually from here or use **Management** from the navigator to enable them all at once, and confirm they have the correct status after startup. Enable any new notification rules as well.

4. Now that you have created and configured the elements and analyses, you will simulate a high latency condition for the Asset Analytics. You may experience high latency overall due to various reasons including

too many analyses, types of inputs, input data density/pattern, expensive functions, analysis scheduling, performance limitations (ie: server resources), etc, or you may experience it on a small number of analyses that have specific issues. It is important to know when there are issues with the analysis service so they can be addressed quickly to prevent problems like gaps in analyses outputs, analyses not updating or old outputs, max latency growing/not recovering, high CPU and/or RAM usage, etc.

a. Select **Elements** in the navigator and go to the **Analyses** tab in the viewer and select the **High Latency** event frame. Click the Expression field for the **HighLatency** start trigger listed in the lower window:



PIAF01 - Asset Analytics

General | Child Elements | Attributes | Ports | Analyses | Notification Rules | Version

Name	Backfilling
Cache Count Missed Percent	
Evaluations Per Second	
High Cache Count Missed Percent	
High Latency	
High Skipped Evaluation Ratio	
Increasing Latency	
Skipped to Good Evaluation Ratio	

Name: High Latency
Description: Detects a high latency for a group of calculations
Categories: Asset Analytics
Analysis Type: ○ Expression ○ Rollup ● Event Frame
Create a new notification rule for High Latency

Generation Mode: Explicit Trigger
Event Frame Template: PSM High Latency Event

Name	Expression	True for	Seve
Variables			

NoRecalculationActive	if BadVal('PI Analysis Service_Recalculation Requests Queued')
	then Exit()
	else ('PI Analysis Service_Recalculation Requests Queued' = 0)		

MaxLatency	if BadVal('PI Analysis Service_Maximum Latency')
	then Exit()
	else 'PI Analysis Service_Maximum Latency'		

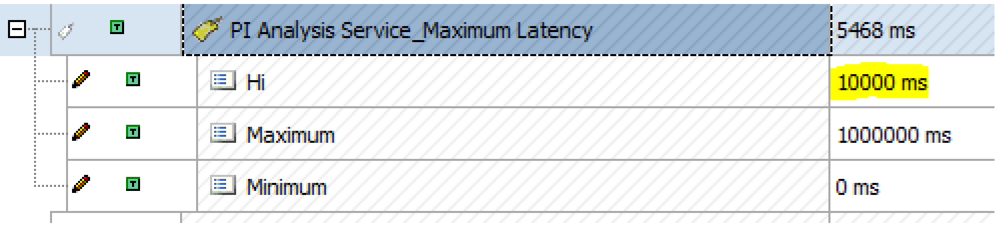
Start triggers			
HighLatency	// During recalculations it is expected that high latency will occur// MaxLatency >'PI Analysis Service_Maximum Latency|Hi' and NoRecalculationActive	Not Set	War
SustainedHigh_Latency	MaxLatency >'PI Analysis Service_Maximum Latency|Hi' and NoRecalculationActive	20 minutes	Maj

```
// During recalculations it is expected that high latency will occur//
MaxLatency >'PI Analysis Service_Maximum Latency|Hi' and NoRecalculationActive
```

i. **HighLatency** - triggers if the **PI Analysis Service_Maximum Latency** is greater than its **Hi** child attribute, and recalculation is not active (because we expect possible high latency during recalc)

ii. There is a second start trigger for **SustainedHigh_Latency**, with the same logic but a **True for = 20 mins**, which will trigger if the latency is sustained for this long

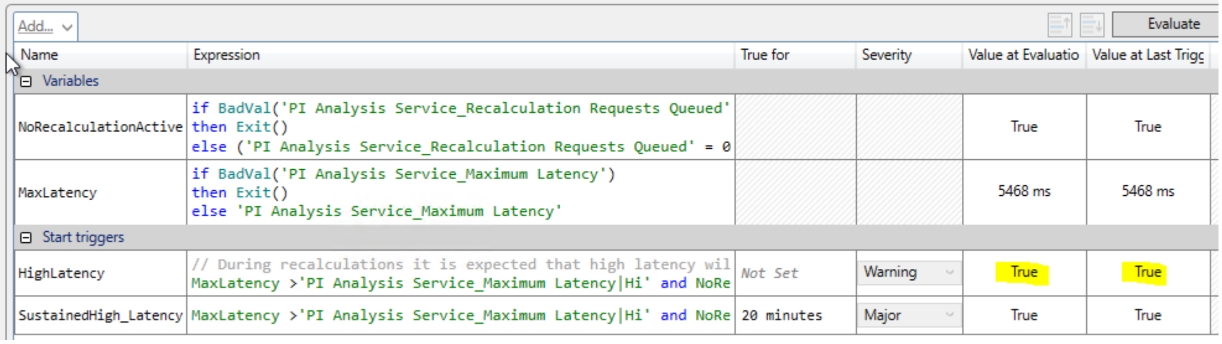
b. Select the **Attributes** tab in the viewer, and find the **PI Analysis Service_Maximum Latency** attribute. Expand the attribute to see the **Hi** child attribute. The threshold value is set to 10000 ms (10 s).



Name	Value
PI Analysis Service_Maximum Latency	5468 ms
Hi	10000 ms
Maximum	1000000 ms
Minimum	0 ms

i. Looking at the **PI Analysis Service_Maximum Latency** value it is below the threshold so set the **Hi** attribute to a lower value (**100** is good). Check in your changes.

c. Select the **Analyses** tab in the Viewer and select the **High Latency** event frame analysis. Click **Evaluate** and see that the value for the **HighLatency** trigger is **True**.



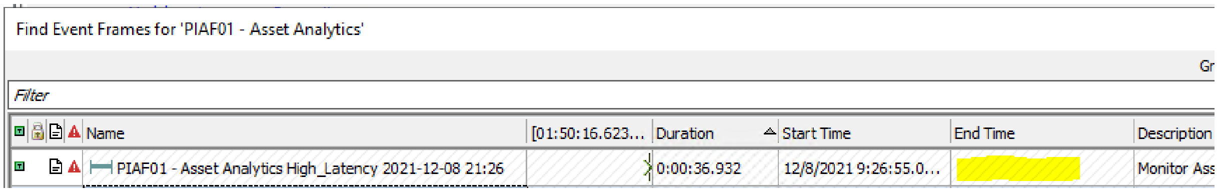
Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
⊟ Variables					

NoRecalculationActive	if BadVal('PI Analysis Service_Recalculation Requests Queued'
	then Exit()
	else ('PI Analysis Service_Recalculation Requests Queued' = 0
				True	True

MaxLatency	if BadVal('PI Analysis Service_Maximum Latency')
	then Exit()
	else 'PI Analysis Service_Maximum Latency'
				5468 ms	5468 ms

⊟ Start triggers					
HighLatency	// During recalculations it is expected that high latency wil MaxLatency >'PI Analysis Service_Maximum Latency|Hi' and NoRe	Not Set	Warning	True	True
SustainedHigh_Latency	MaxLatency >'PI Analysis Service_Maximum Latency|Hi' and NoRe	20 minutes	Major	True	True

d. Select the **General** tab and click **Event Frames** – you will see a new event frame for **Asset Analytics High_Latency**. The blank **End Time** shows the event frame is still active.

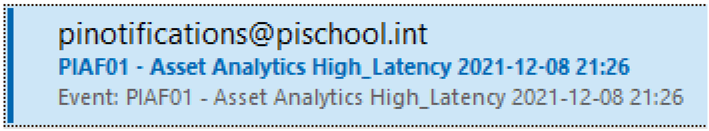


Find Event Frames for 'PIAF01 - Asset Analytics'

Name	[01:50:16.623...	Duration	Start Time	End Time	Description
PIAF01 - Asset Analytics High_Latency 2021-12-08 21:26		0:00:36.932	12/8/2021 9:26:55.0...		Monitor Ass

i. **Close** the **Event Frame** window.

e. Open Microsoft Outlook on the **PIMONITOR01** machine. In the *student01@pischool.int* Inbox locate the email for the **Asset Analytics High_Latency** notification. Refresh if it is not showing.



pinotifications@pischool.int
PIAF01 - Asset Analytics High_Latency 2021-12-08 21:26
Event: PIAF01 - Asset Analytics High_Latency 2021-12-08 21:26

i. Open the email and look at the content.

f. Select the **Attributes** tab in the viewer and change the value for **PI Analysis Service_Maximum Latency|Hi** back to 10000 ms. Check in your changes.

i. Select the **Analyses** tab in the Viewer and select the **High Latency** event frame analysis. Click **Evaluate** and see that the value for the **HighLatency** trigger is now **False**.

ii. Select the **General** tab and click **Event Frames**. Confirm the event frame has closed – it will have an **End Time** now.

iii. You should also get a **Notification is closed** email.

8. Optional exercise 1: Server – high processor utilization

Exercise objectives

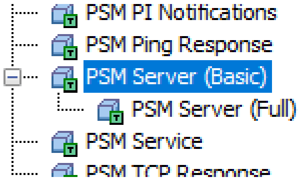
- Configure the elements in your AF structure for monitoring a general server (you will use a Data Archive)
- Create required monitoring tags
- Understand the analyses used for monitoring
- Simulate a high processor utilization condition

Scenario

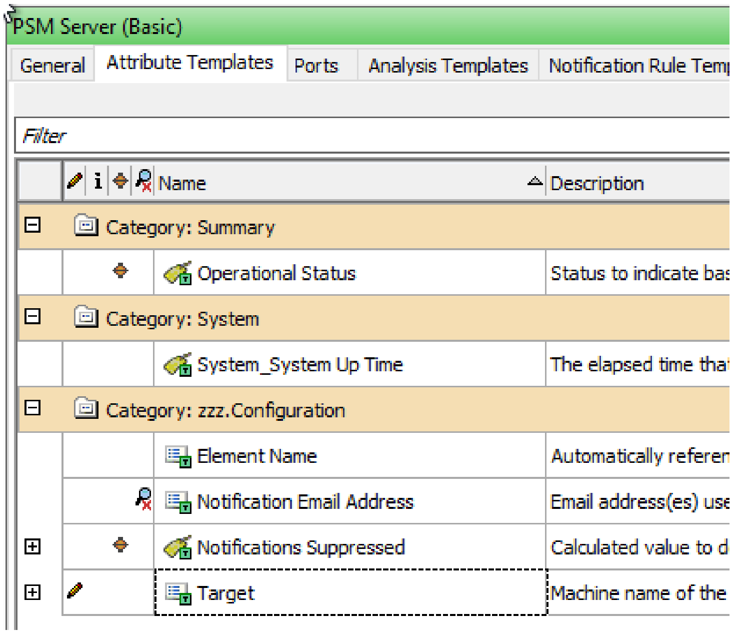
The element necessary for general server monitoring already exists (you created it in exercise 2). You will review the monitoring provided in the existing element, and then change the template used for this element from PSM Server (Basic) to PSM Server (Full). You will review the new attributes and analyses and determine how they are used for monitoring, and what triggers them. You will simulate a high processor utilization condition by triggering an analysis to produce an event frame and notification, and review these outputs.

Approach

1. Select **Library** in the navigator, right-click on **Element Templates** and select **Arrange by > Arrange By Template Inheritance**. This allows us to see any derived templates.
    a. The **PSM Server (*)** templates are a set of derived templates. Notice the **+** next to **PSM Server (Basic)** – if you expand that you will see **PSM Server (Full)**.

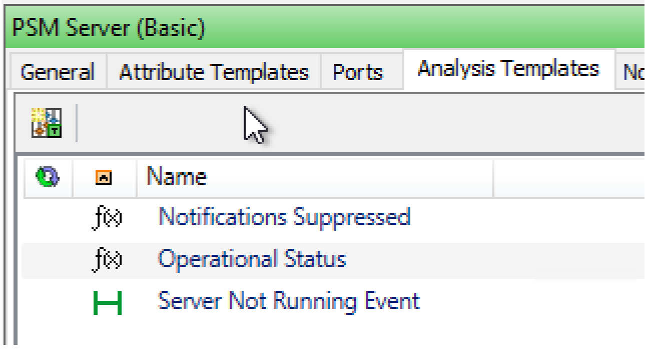


b. Select **PSM Server (Basic)** template from the tree. On the **Attribute Templates** tab in the viewer, we can see the handful of configuration attributes it contains



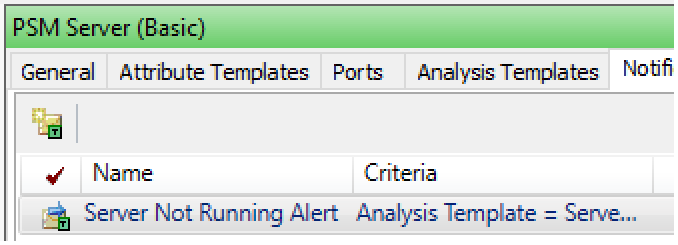
Name	Description
Category: Summary	
Operational Status	Status to indicate bas
Category: System	
System_System Up Time	The elapsed time that
Category: zzz.Configuration	
Element Name	Automatically referen
Notification Email Address	Email address(es) use
Notifications Suppressed	Calculated value to d
Target	Machine name of the

i. On the **Analysis Templates** tab in the viewer, we can see the three analyses it contains:



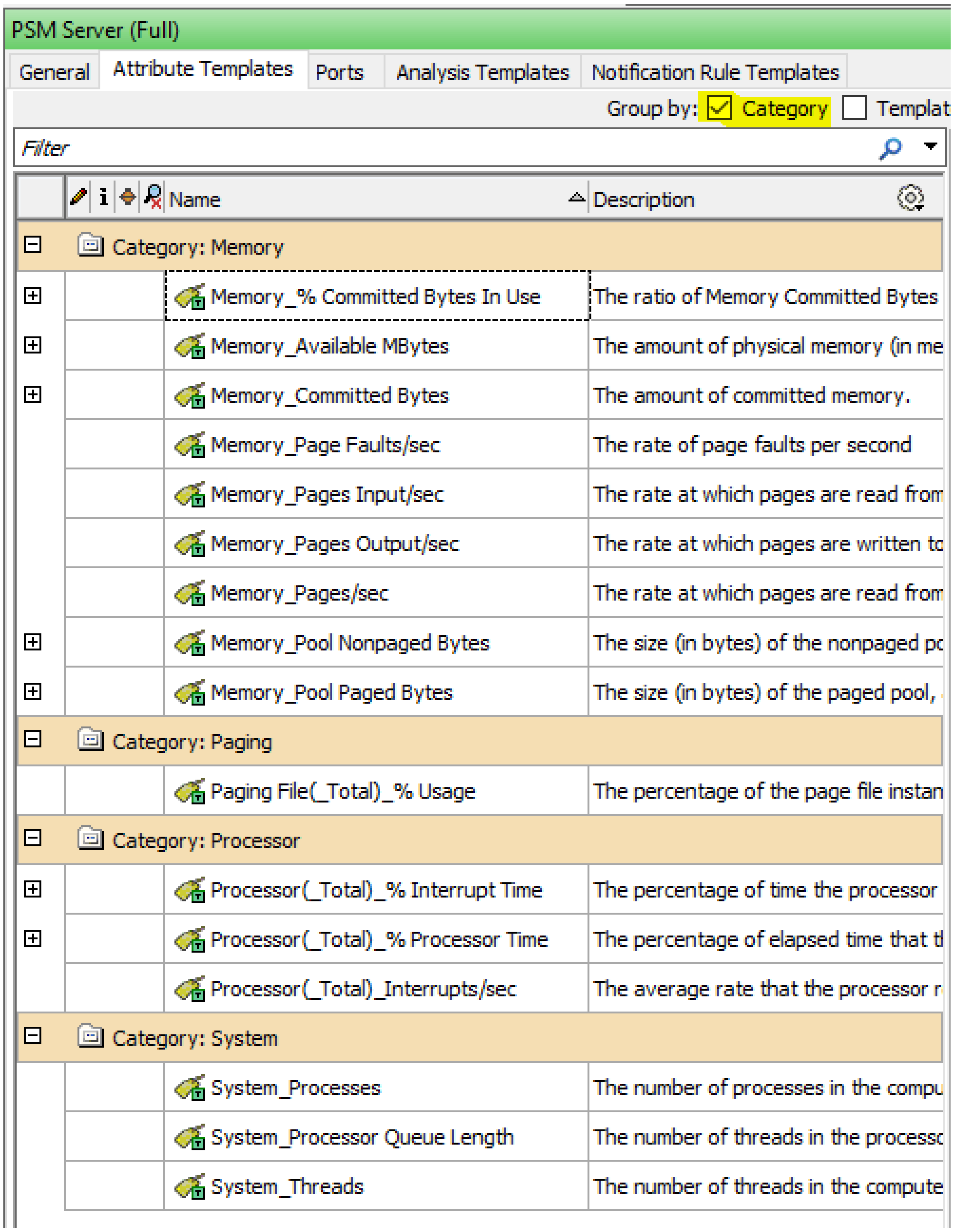
Name
Notifications Suppressed
Operational Status
Server Not Running Event

ii. On the **Notification Rule Templates** tab we can its single notification rule:



Name	Criteria
Server Not Running Alert	Analysis Template = Serve...

c. The **PSM Server (Full)** template is derived from **PSM Server (Basic)**. Select it, and notice additional attributes have been added on the **Attribute Templates** tab. Note that you will only see the attributes specific to this template if you have **Group by Category** selected (see below).

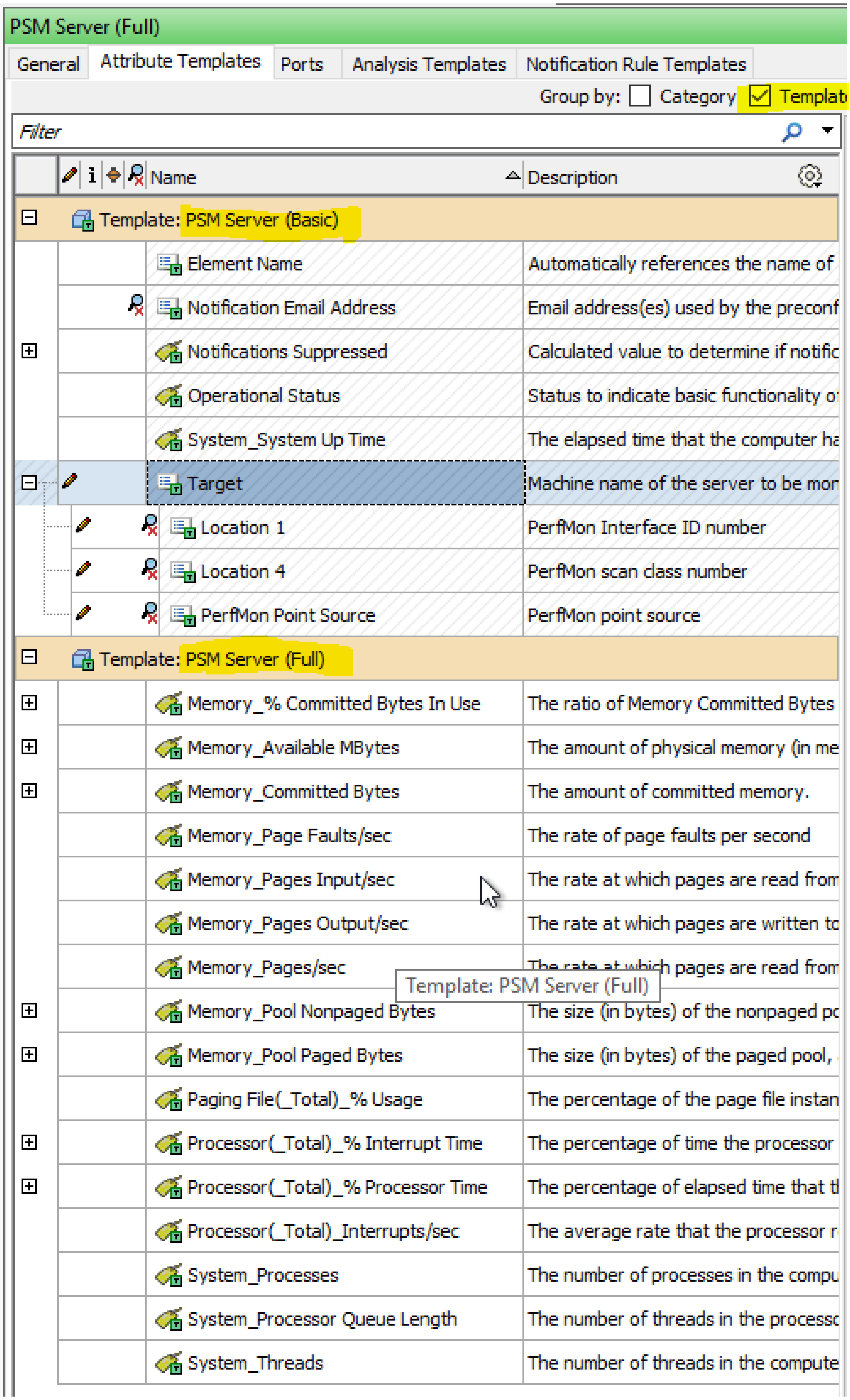


PSM Server (Full)

Group by: ☑ Category □ Templat

	Name	Description
⊟	Category: Memory	
⊞	Memory_% Committed Bytes In Use	The ratio of Memory Committed Bytes
⊞	Memory_Available MBytes	The amount of physical memory (in me
⊞	Memory_Committed Bytes	The amount of committed memory.
	Memory_Page Faults/sec	The rate of page faults per second
	Memory_Pages Input/sec	The rate at which pages are read from
	Memory_Pages Output/sec	The rate at which pages are written to
	Memory_Pages/sec	The rate at which pages are read from
⊞	Memory_Pool Nonpaged Bytes	The size (in bytes) of the nonpaged po
⊞	Memory_Pool Paged Bytes	The size (in bytes) of the paged pool,
⊟	Category: Paging	
	Paging File(_Total)_% Usage	The percentage of the page file instan
⊟	Category: Processor	
⊞	Processor(_Total)_% Interrupt Time	The percentage of time the processor
⊞	Processor(_Total)_% Processor Time	The percentage of elapsed time that th
	Processor(_Total)_Interrupts/sec	The average rate that the processor r
⊟	Category: System	
	System_Processes	The number of processes in the compu
	System_Processor Queue Length	The number of threads in the processo
	System_Threads	The number of threads in the compute

To see the attributes for this template along with those inherited from **PSM Server (Basic)** (such as Target and other attributes), select **Group by Template**.

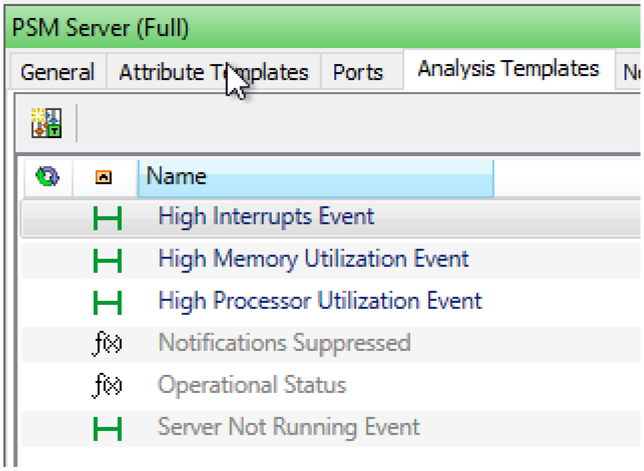


PSM Server (Full)

Group by: □ Category ☑ Templat

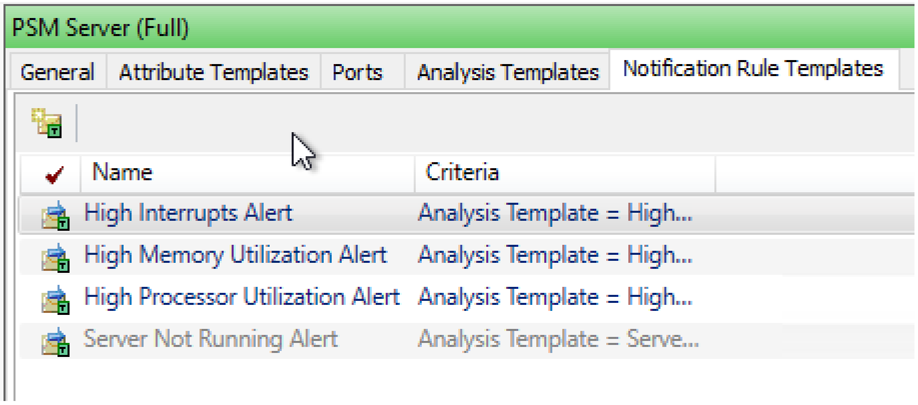
Name	Description
⊟ Template: PSM Server (Basic)	
Element Name	Automatically references the name of
Notification Email Address	Email address(es) used by the preconf
⊞ Notifications Suppressed	Calculated value to determine if notific
Operational Status	Status to indicate basic functionality o
System_System Up Time	The elapsed time that the computer ha
⊟ Target	Machine name of the server to be mon
Location 1	PerfMon Interface ID number
Location 4	PerfMon scan class number
PerfMon Point Source	PerfMon point source
⊟ Template: PSM Server (Full)	
⊞ Memory_% Committed Bytes In Use	The ratio of Memory Committed Bytes
⊞ Memory_Available MBytes	The amount of physical memory (in me
⊞ Memory_Committed Bytes	The amount of committed memory.
Memory_Page Faults/sec	The rate of page faults per second
Memory_Pages Input/sec	The rate at which pages are read from
Memory_Pages Output/sec	The rate at which pages are written to
Memory_Pages/sec	The rate at which pages are read from
⊞ Memory_Pool Nonpaged Bytes	The size (in bytes) of the nonpaged po
⊞ Memory_Pool Paged Bytes	The size (in bytes) of the paged pool,
Paging File(_Total)_% Usage	The percentage of the page file instan
⊞ Processor(_Total)_% Interrupt Time	The percentage of time the processor
⊞ Processor(_Total)_% Processor Time	The percentage of elapsed time that th
Processor(_Total)_Interrupts/sec	The average rate that the processor r
System_Processes	The number of processes in the compu
System_Processor Queue Length	The number of threads in the processo
System_Threads	The number of threads in the compute

i. Select the **Analysis Templates** tab and you will see analyses for **Notifications Suppressed, Operational Status, and Server Not Running Event** which are greyed out. Those analyses have been inherited from **PSM Server (Basic).** New analyses added on this template include High Interrupts Event, **High Memory Utilization Event**, and **High Processor Utilization Event**.



Name
High Interrupts Event
High Memory Utilization Event
High Processor Utilization Event
Notifications Suppressed
Operational Status
Server Not Running Event

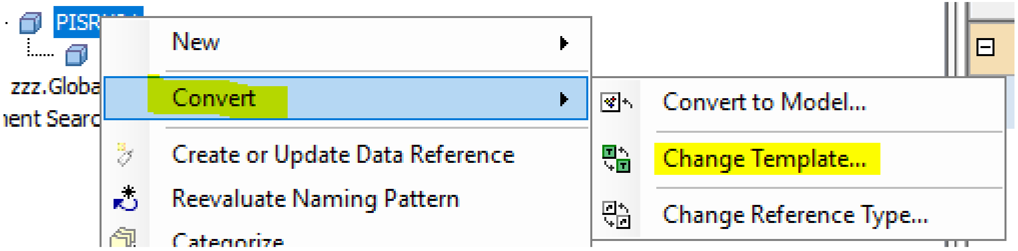
ii. The new and inherited notification rules are seen on the **Notification Rule Templates** tab.



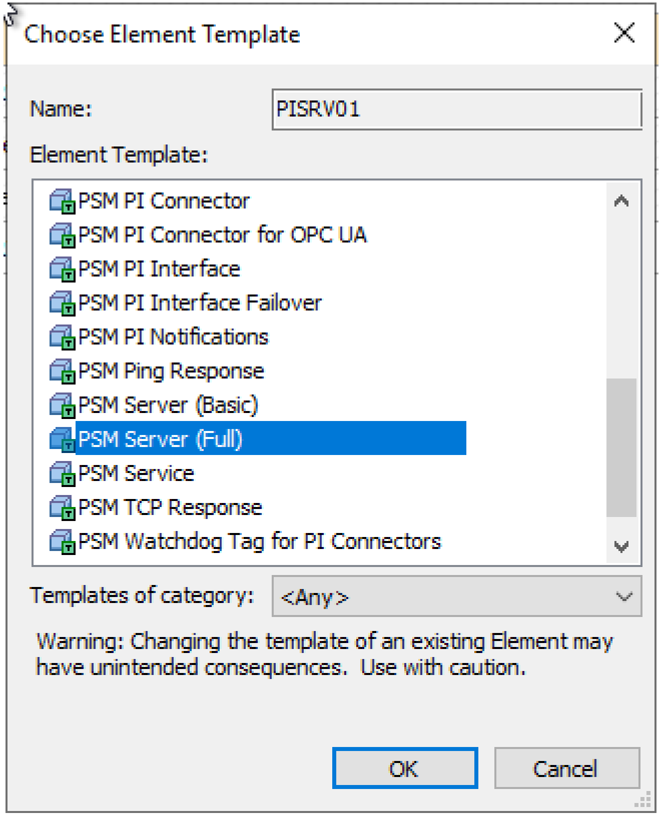
Name	Criteria
High Interrupts Alert	Analysis Template = High...
High Memory Utilization Alert	Analysis Template = High...
High Processor Utilization Alert	Analysis Template = High...
Server Not Running Alert	Analysis Template = Serve...

iii. So as you can see, the derived templates build upon existing components on the parent template. This allows you to use either of the Server templates to create your server elements, depending on what level of monitoring you require. It also makes it easy to create a hierarchy quickly using the **PSM Server (Basic)** template, and later change to the **PSM Server (Full)** template if you want additional monitoring.

d. Select **Elements** in the navigator, and in the browser navigate to the **PISRV01** (server) element. Rt-click, and select **Convert > Change Template...**



i. In the Choose Element Template window select **PSM Server (Full)** and click **OK**.



Choose Element Template

Name: PISRV01

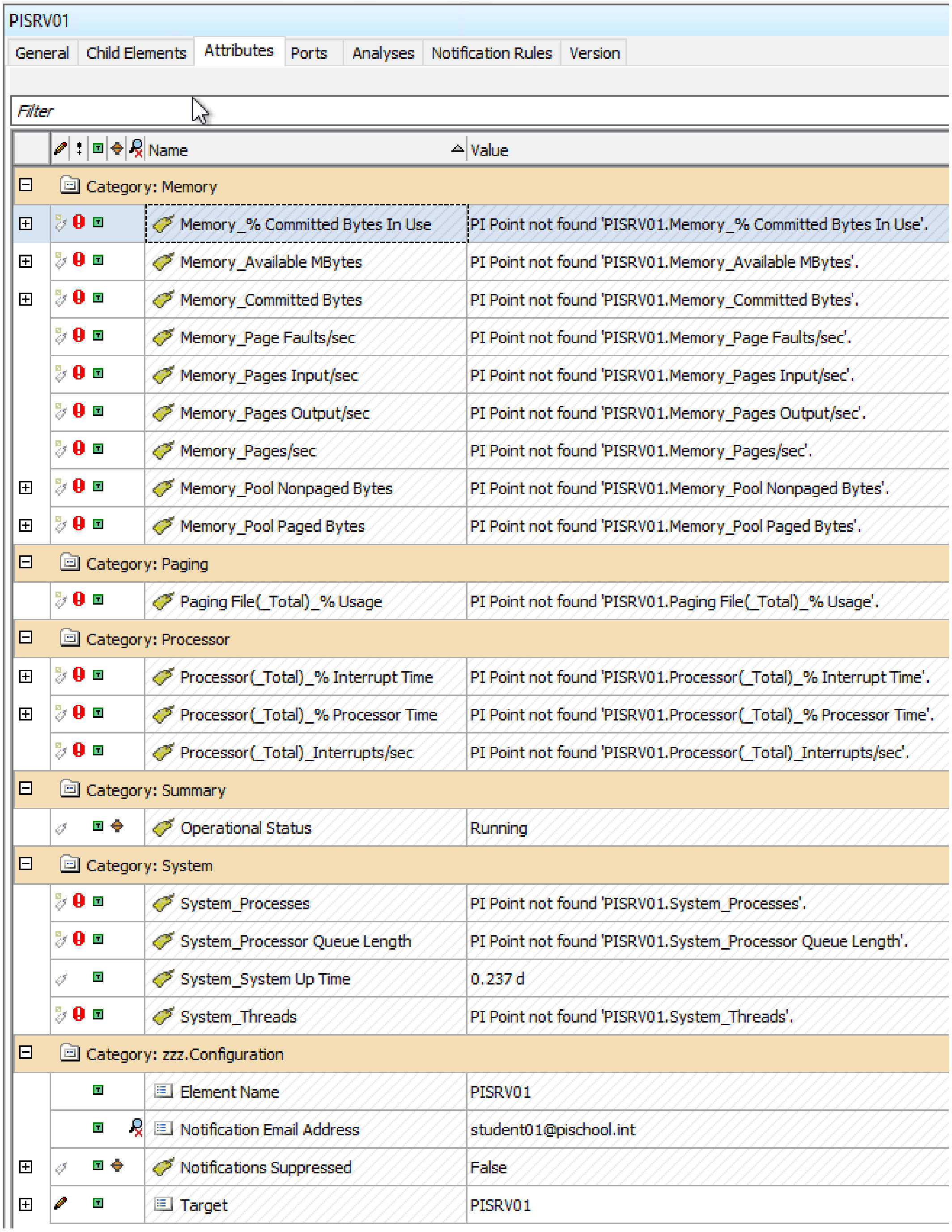
Element Template:

- PSM PI Connector
- PSM PI Connector for OPC UA
- PSM PI Interface
- PSM PI Interface Failover
- PSM PI Notifications
- PSM Ping Response
- PSM Server (Basic)
- PSM Server (Full)
- PSM Service
- PSM TCP Response
- PSM Watchdog Tag for PI Connectors

Templates of category: <Any>

Warning: Changing the template of an existing Element may have unintended consequences. Use with caution.

ii. On the Attributes tab in the viewer, notice that where you previously only had a handful of attributes, there are now many.



Name	Value
Category: Memory	
Memory_% Committed Bytes In Use	PI Point not found 'PISRV01.Memory_% Committed Bytes In Use'.
Memory_Available MBytes	PI Point not found 'PISRV01.Memory_Available MBytes'.
Memory_Committed Bytes	PI Point not found 'PISRV01.Memory_Committed Bytes'.
Memory_Page Faults/sec	PI Point not found 'PISRV01.Memory_Page Faults/sec'.
Memory_Pages Input/sec	PI Point not found 'PISRV01.Memory_Pages Input/sec'.
Memory_Pages Output/sec	PI Point not found 'PISRV01.Memory_Pages Output/sec'.
Memory_Pages/sec	PI Point not found 'PISRV01.Memory_Pages/sec'.
Memory_Pool Nonpaged Bytes	PI Point not found 'PISRV01.Memory_Pool Nonpaged Bytes'.
Memory_Pool Paged Bytes	PI Point not found 'PISRV01.Memory_Pool Paged Bytes'.
Category: Paging	
Paging File(_Total)_% Usage	PI Point not found 'PISRV01.Paging File(_Total)_% Usage'.
Category: Processor	
Processor(_Total)_% Interrupt Time	PI Point not found 'PISRV01.Processor(_Total)_% Interrupt Time'.
Processor(_Total)_% Processor Time	PI Point not found 'PISRV01.Processor(_Total)_% Processor Time'.
Processor(_Total)_Interrupts/sec	PI Point not found 'PISRV01.Processor(_Total)_Interrupts/sec'.
Category: Summary	
Operational Status	Running
Category: System	
System_Processes	PI Point not found 'PISRV01.System_Processes'.
System_Processor Queue Length	PI Point not found 'PISRV01.System_Processor Queue Length'.
System_System Up Time	0.237 d
System_Threads	PI Point not found 'PISRV01.System_Threads'.
Category: zzz.Configuration	
Element Name	PISRV01
Notification Email Address	student01@pischool.int
Notifications Suppressed	False
Target	PISRV01

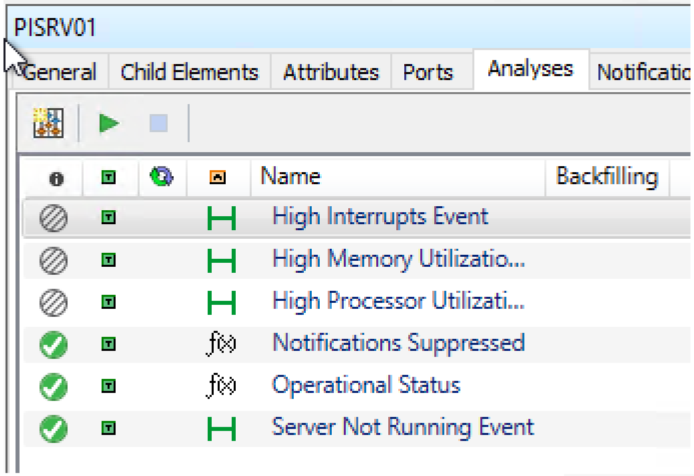
iii. Since you had already populated the configuration items when you created the element, you shouldn't need to do anything else there, but it's always good to check when you change templates, in case additional configuration items were added by a derived template.

iv. Check in your changes.

e. Right-click on the **PISRV01** element and select **Create or Update Data Reference** to create the PI Tags for all the new attributes provided by the **PSM Server (Full)** template.

   i. Once these have been created, the values will show as **Pt Created** until they get picked up by the PIPerfMon interface and populated. Wait a few minutes and click Refresh to confirm the PI Tags are being populated.

f. Select the **Analyses** tab in the viewer and you will see the three new event frame analyses defined for this element which are disabled, along with the existing 3 analyses which are running.



PISRV01

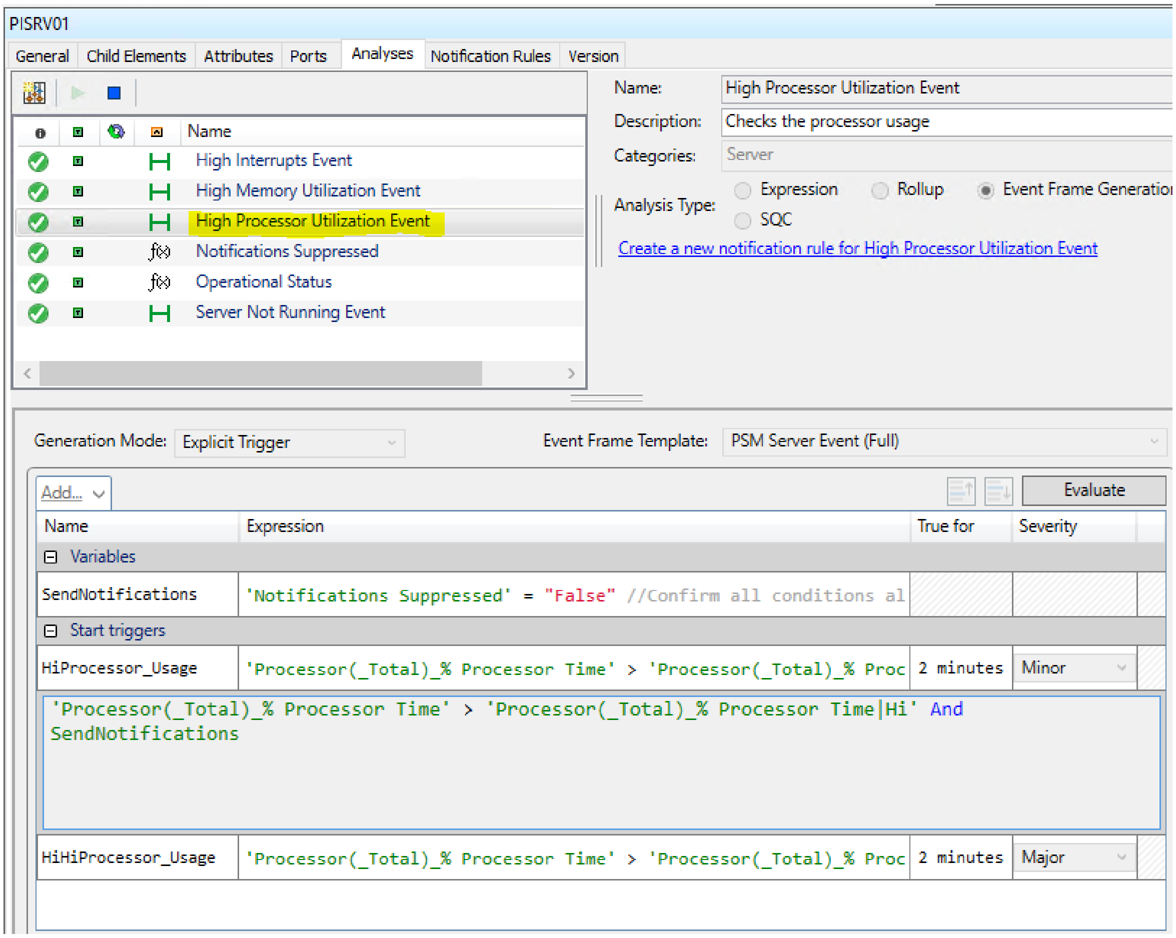
General | Child Elements | Attributes | Ports | Analyses | Notificatio

	Name	Backfilling
▨	High Interrupts Event	
▨	High Memory Utilizatio...	
▨	High Processor Utilizati...	
✓	Notifications Suppressed	
✓	Operational Status	
✓	Server Not Running Event	

   i. Enable the analyses individually or use **Management** from the navigator and confirm they have the correct status after startup. Enable any new notification rules as well.

2. Now that you have configured the element and analyses, you will simulate a high processor utilization condition. Servers generally have several processes running at any given time, and they compete for resources. It's important to monitor the overall processor usage on your server to ensure it isn't being overloaded.

a. Back in the **Element** view, select the **Analyses** tab in the viewer and select the **High Processor Utilization** event frame.



PISRV01

Name
High Interrupts Event
High Memory Utilization Event
High Processor Utilization Event
Notifications Suppressed
Operational Status
Server Not Running Event

Name: High Processor Utilization Event

Description: Checks the processor usage

Categories: Server

Analysis Type: ○ Expression ○ Rollup ● Event Frame Generatio ○ SQC

Create a new notification rule for High Processor Utilization Event

Generation Mode: Explicit Trigger

Event Frame Template: PSM Server Event (Full)

Name	Expression	True for	Severity
⊟ Variables			
SendNotifications	'Notifications Suppressed' = "False" //Confirm all conditions al		
⊟ Start triggers			
HiProcessor_Usage	'Processor(_Total)_% Processor Time' > 'Processor(_Total)_% Proc	2 minutes	Minor
HiHiProcessor_Usage	'Processor(_Total)_% Processor Time' > 'Processor(_Total)_% Proc	2 minutes	Major

```
'Processor(_Total)_% Processor Time' > 'Processor(_Total)_% Processor Time|Hi' And
SendNotifications
```

There are one variable and two start triggers defined for this event frame:

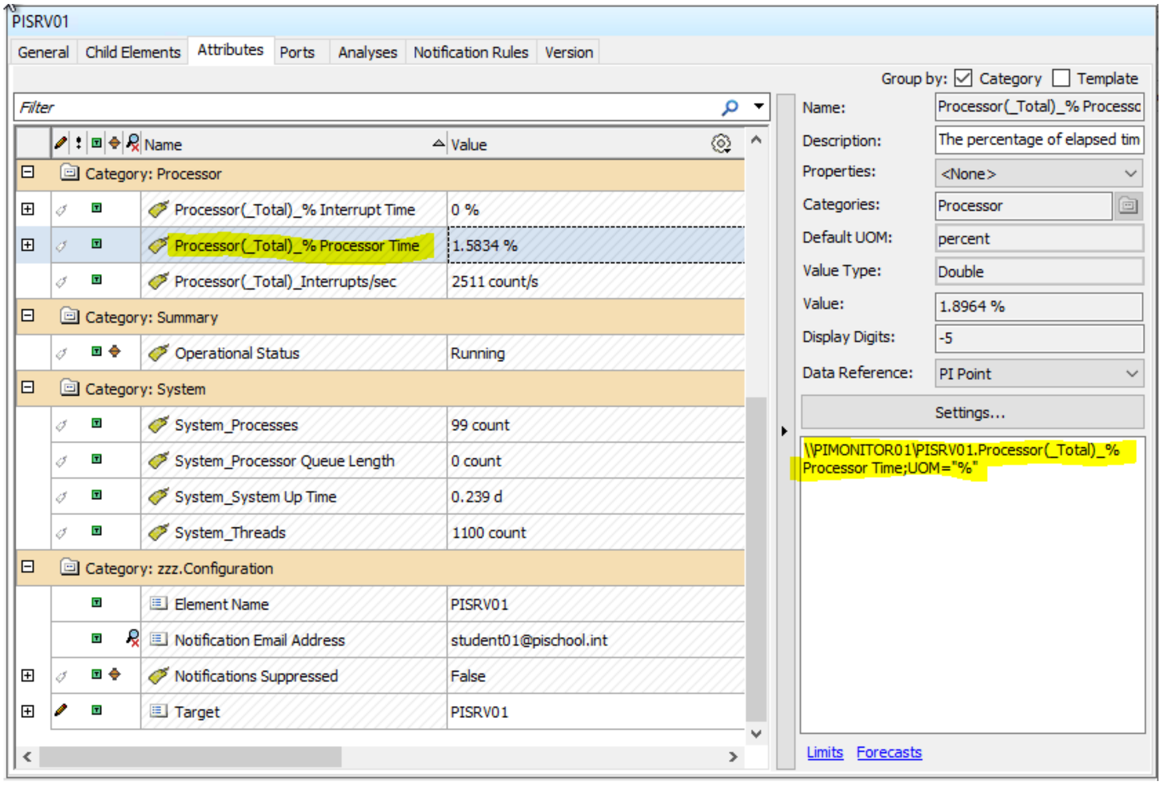
i. Variable **SendNotifications** is checking conditions to see if it should be sending notifications:

1. Have notifications been suppressed - the server's **Notifications Suppressed** attribute would be **True** if it had been suppressed, and **False** otherwise. See *Notification suppression* for more information.

ii. Trigger **Hi_Processor_Usage** – compares the value of the **Processor(_Total)_% Processor Time** attribute to the **Processor(_Total)_% Processor Time|Hi** child attribute. If the attribute value is greater than the **Hi** threshold for 2 minutes, this event frame will be triggered.

iii. Similarly **HiHi_Processor_Usage** – does the same comparison but to the **Processor(_Total)_% Processor Time|HiHi** child attribute.

b. Select the **Attributes** tab in the viewer and find the attribute the analyses was evaluating (**Processor(_Total)_% Processor Time**). Select the attribute and note that it is a PI Point data reference with a unit of measure of percent, with the associated PI tag listed under **Settings** on the right.



Name	Value
Category: Processor	
Processor(_Total)_% Interrupt Time	0 %
Processor(_Total)_% Processor Time	1.5834 %
Processor(_Total)_Interrupts/sec	2511 count/s
Category: Summary	
Operational Status	Running
Category: System	
System_Processes	99 count
System_Processor Queue Length	0 count
System_System Up Time	0.239 d
System_Threads	1100 count
Category: zzz.Configuration	
Element Name	PISRV01
Notification Email Address	student01@pischool.int
Notifications Suppressed	False
Target	PISRV01

Name: Processor(_Total)_% Processc

Description: The percentage of elapsed tim

Properties: <None>

Categories: Processor

Default UOM: percent

Value Type: Double

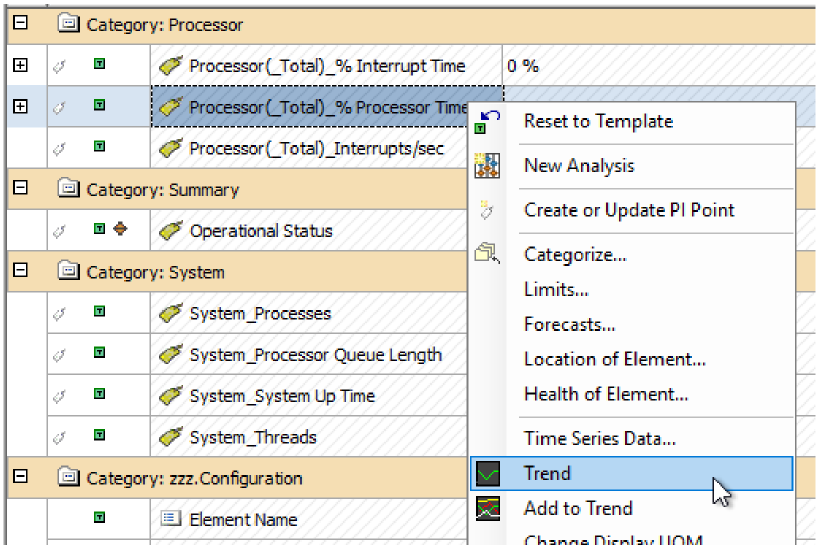
Value: 1.8964 %

Display Digits: -5

Data Reference: PI Point

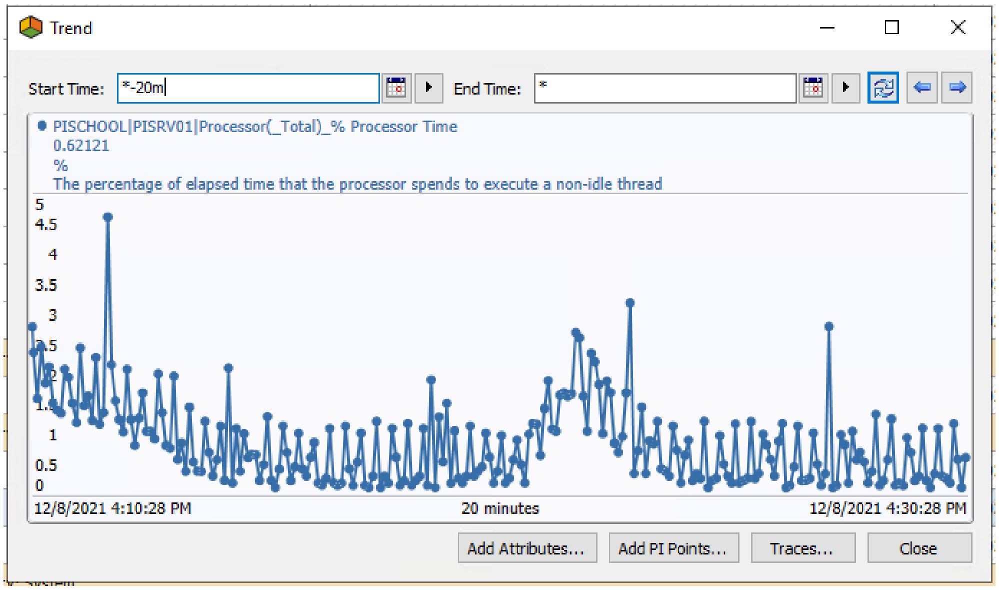
\\PIMONITOR01\PISRV01.Processor(_Total)_% Processor Time;UOM="%"

i. Right-click the attribute and select **Trend**:



Name	Value
Category: Processor	
Processor(_Total)_% Interrupt Time	0 %
Processor(_Total)_% Processor Time	
Processor(_Total)_Interrupts/sec	
Category: Summary	
Operational Status	
Category: System	
System_Processes	
System_Processor Queue Length	
System_System Up Time	
System_Threads	
Category: zzz.Configuration	
Element Name	

ii. This will open a trend of the tag value in a popup window. If needed, set the start time and end time and hit **Enter** to see the values. For the system below you can see that the value tends to vary between 0.25 and about 1.5%, with spikes up to 4% - note the characteristics of your own system.



Trend

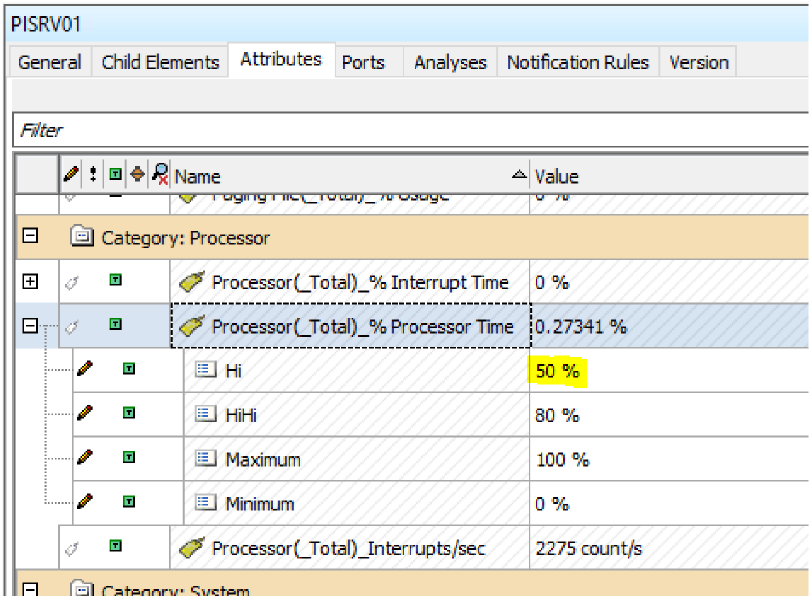
Start Time: *-20m

End Time: *

PISCHOOL|PISRV01|Processor(_Total)_% Processor Time
0.62121
%
The percentage of elapsed time that the processor spends to execute a non-idle thread

iii. Close the trend popup window.

c. Expand the attribute to see the child attributes. The threshold value for **Hi** is set to **50%**.



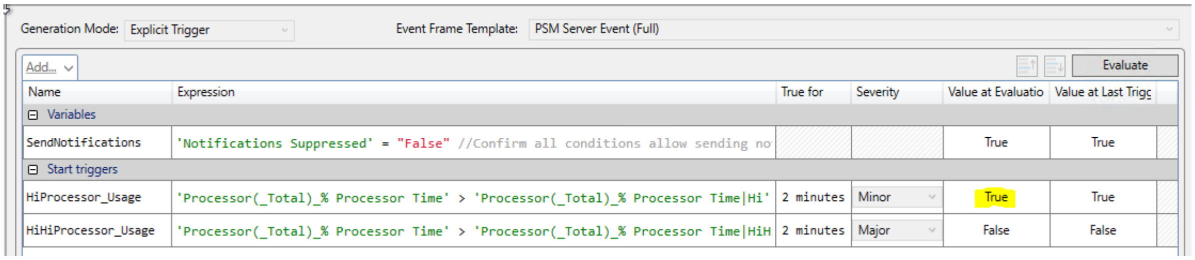
PISRV01

Name	Value
Paging File(_Total)_% Usage	0 %
Category: Processor	
Processor(_Total)_% Interrupt Time	0 %
Processor(_Total)_% Processor Time	0.27341 %
Hi	50 %
HiHi	80 %
Maximum	100 %
Minimum	0 %
Processor(_Total)_Interrupts/sec	2275 count/s
Category: System	

i. Remember that the triggering condition is when the **% Processor Time** is higher than the **Hi** threshold for 2 minutes. Change the value for **Hi** to **0**.

ii. Ensure you check in your changes.

d. Go to the **Analyses** tab and select the **High Processor Utilization** analysis. Click **Evaluate** at the top right of the lower pane to confirm if the trigger conditions are being met – it will say **True** under **Value at Evaluation** if the trigger is met. You will have to wait for it to be true for 2 minutes until the event frame gets created. You can see how long it's been since the event last triggered by looking at the bottom of the viewer at the **Last Trigger Time.**

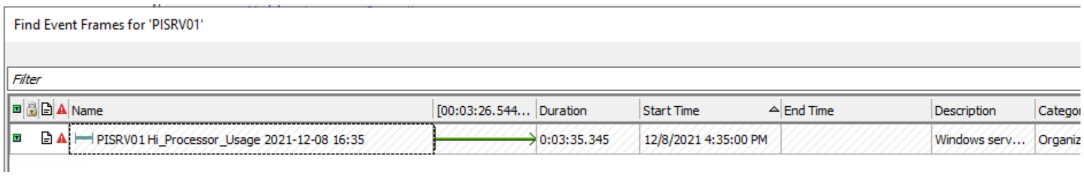


Generation Mode: Explicit Trigger   Event Frame Template: PSM Server Event (Full)

Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigc
⊟ Variables					
SendNotifications	'Notifications Suppressed' = "False" //Confirm all conditions allow sending no			True	True
⊟ Start triggers					
HiProcessor_Usage	'Processor(_Total)_% Processor Time' > 'Processor(_Total)_% Processor Time|Hi'	2 minutes	Minor	True	True
HiHiProcessor_Usage	'Processor(_Total)_% Processor Time' > 'Processor(_Total)_% Processor Time|HiH	2 minutes	Major	False	False

i. Click on the **General** tab in the Viewer, and click the **Event Frames** link to see any event frames created by this element.

ii. Once the new event frame has been created it will show in the **Find Event Frames** window. You can see it is currently active (not yet closed) because the **End Time** is blank.

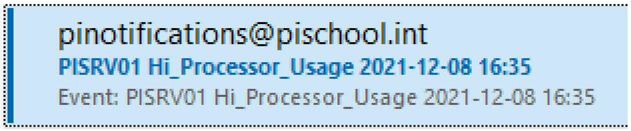


Find Event Frames for 'PISRV01'

Name	[00:03:26.544...	Duration	Start Time	End Time	Description	Categoi
PISRV01 Hi_Processor_Usage 2021-12-08 16:35		0:03:35.345	12/8/2021 4:35:00 PM		Windows serv...	Organiz

e. **Close** the **Find Event Frame** window. Open Microsoft Outlook on the PIMONITOR01 machine.

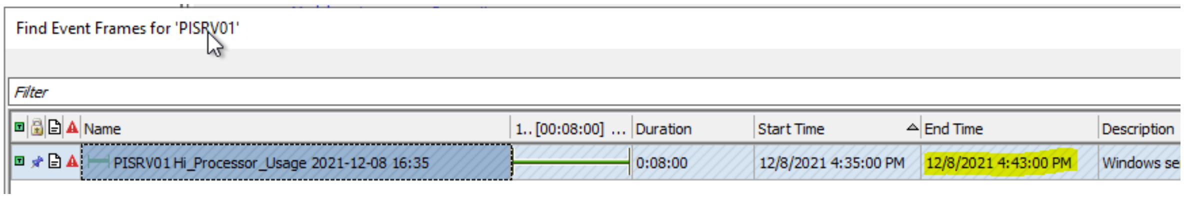
i. Select the *student01@pischool.int* Inbox and locate the email notification for the **High Processor Usage** notification. If it is not showing yet, click **Send/Receive All Folders** on the top right of the **Home** Toolbar to refresh.



pinotifications@pischool.int
PISRV01 Hi_Processor_Usage 2021-12-08 16:35
Event: PISRV01 Hi_Processor_Usage 2021-12-08 16:35

ii. Open the email and look at the content. The event name (which is also the email subject) shows which condition triggered the event. The table provides some key information on performance indicators. Below the table are instructions to troubleshoot the issue and a link to the event details view in PI Vision.

f. Close the event frame by changing the **Hi** attribute value back to **50%** and checking in your changes. Click **Event Frames** on the **General** tab to confirm the event frame has closed (the **End Time** will be populated)



Find Event Frames for 'PISRV01'

Name	1.. [00:08:00] ...	Duration	Start Time	End Time	Description
PISRV01 Hi_Processor_Usage 2021-12-08 16:35		0:08:00	12/8/2021 4:35:00 PM	12/8/2021 4:43:00 PM	Windows se

g. You should also receive a Notification closed email.

9. Optional exercise 2: PI Buffer subsystem – low buffer queue capacity

Exercise objectives

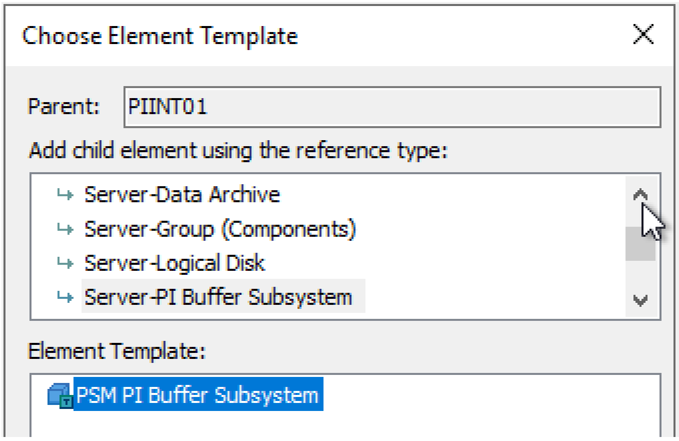
- Create and configure the elements in your AF structure for monitoring buffering on a PI System interface node
- Create required monitoring tags
- Understand the analyses used for monitoring
- Simulate low buffer queue capacity scenario

Scenario

You will create the elements necessary for monitoring buffering on a PI System interface node. After reviewing what configuration data is required for the elements to function, you will locate the required data and configure the elements appropriately. You will review the analyses and determine how they are used for monitoring, and what triggers them. You will simulate a low buffer queue capacity scenario by triggering analyses to produce event frames and notifications, and review these outputs.

Approach

1. Now you will create the element for the interface node buffering monitoring. This element goes at the same level as the interface instance element, under the interface machine. Buffering is used with PI Interfaces to prevent data loss when the Data Archive is unavailable. When the interface has data to send to the Data Archive, the data first flows into the buffer queues on the interface node, and then to the Data Archive. If the buffer subsystem on the interface node cannot connect to the Data Archive to send data, it keeps queuing data until the Data Archive becomes available, and then flushes the queues to the Data Archive.
    a. In PI System Explorer, in the browser right-click the **PIINT01** element in your database and select **New Child Element**.
    b. In the Choose Element Template window, ensure reference type **Server-PI Buffer Subsystem** is chosen, and select the **PSM PI Buffer Subsystem** template (Note: buffering can be used by multiple interface types so this template is generic):



Choose Element Template

Parent: PIINT01

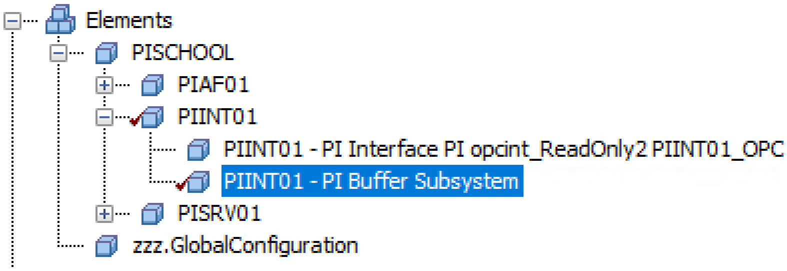
Add child element using the reference type:

- Server-Data Archive
- Server-Group (Components)
- Server-Logical Disk
- Server-PI Buffer Subsystem

Element Template:

PSM PI Buffer Subsystem

i. You will see a new PI Buffer Subsystem element with a default name under PIINT01 in the element structure. This name can stay as-is:



- Elements
  - PISCHOOL
    - PIAF01
    - PIINT01
      - PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
      - PIINT01 - PI Buffer Subsystem
    - PISRV01
  - zzz.GlobalConfiguration

ii. Check in your changes.

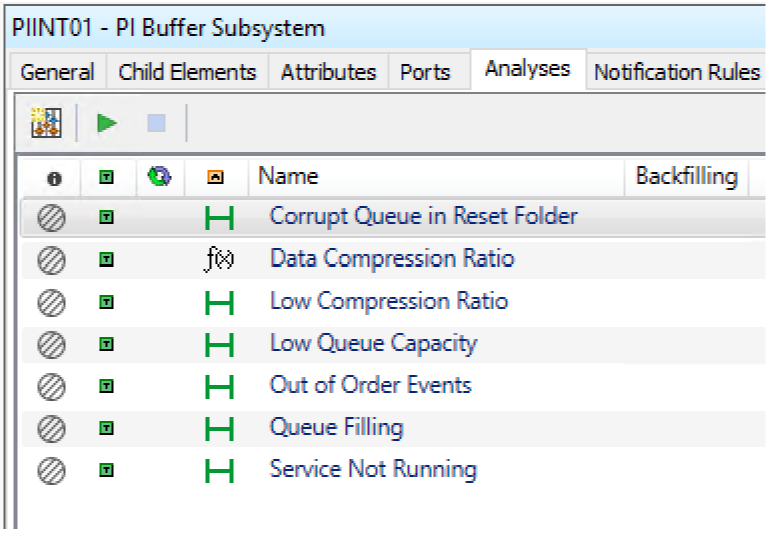
c. Select the **Attributes** tab in the viewer, and confirm the configuration attribute values were inherited properly from the parent element.

2. There are several attributes which are PI Point data references. You will need to create the tags for these.

a. Right-click on the **PIINT01 – PI Buffer Subsystem** element in the browser and select **Create or Update Data Reference**. The Create Config window will show the progress of the creation.

b. Once the tags have finished creating, you will see a value of **Pt Created**. Wait a few minutes, and click Refresh on the toolbar to see values.

c. Select the **Analyses** tab in the viewer and you will see several analyses defined for this element.



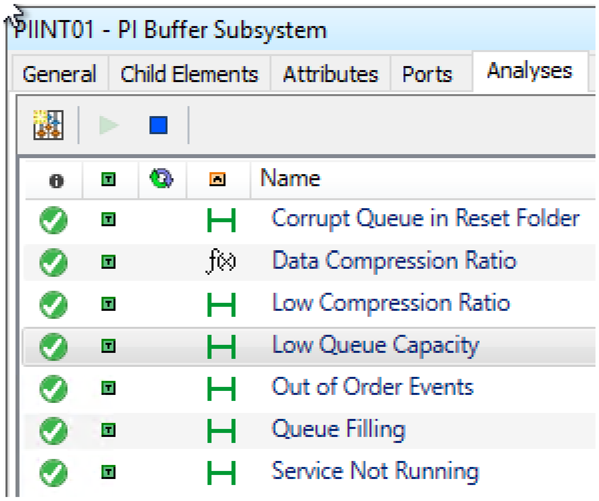
PIINT01 - PI Buffer Subsystem

				Name	Backfilling
▨	🟩		⊢⊣	Corrupt Queue in Reset Folder	
▨	🟩		f(x)	Data Compression Ratio	
▨	🟩		⊢⊣	Low Compression Ratio	
▨	🟩		⊢⊣	Low Queue Capacity	
▨	🟩		⊢⊣	Out of Order Events	
▨	🟩		⊢⊣	Queue Filling	
▨	🟩		⊢⊣	Service Not Running	

d. As previously, the analyses are disabled. Enable them (and the notification rules) via the **Management** pane.

3. Now that you have created and configured the element and analyses, you will simulate a low buffer queue capacity condition. The buffer subsystem requires disk space for the buffer queue files to store data – the performance counters provided for the buffer subsystem analyze how much time it would take to fill up the queue files based on existing data rates and available disk space. It is important to know when you are in danger of running out of space for the queue files as it could result in data loss. You would need to resolve this by adding or freeing up disk space, or lowering your data rate.

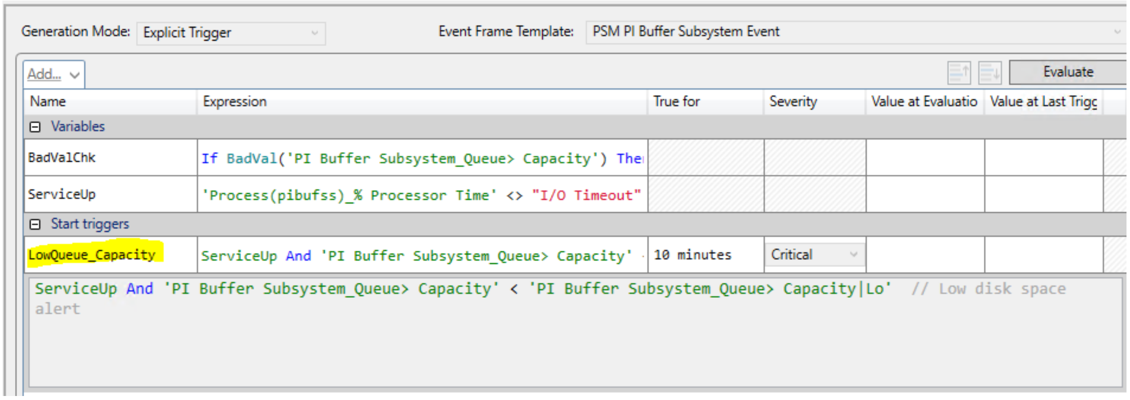
a. Select the **Analyses** tab in the viewer and select the **Low Queue Capacity** event frame.



PIINT01 - PI Buffer Subsystem

Name
Corrupt Queue in Reset Folder
Data Compression Ratio
Low Compression Ratio
Low Queue Capacity
Out of Order Events
Queue Filling
Service Not Running

i. This analysis is configured with one start trigger. Click in the expression field for that trigger to see the full expression:



Generation Mode: Explicit Trigger

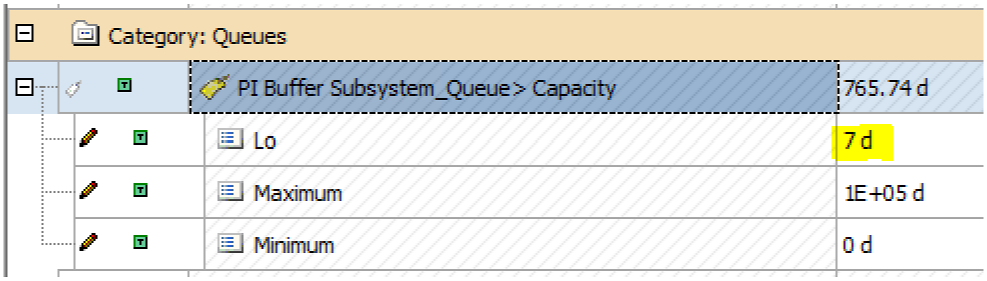
Event Frame Template: PSM PI Buffer Subsystem Event

Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigc
⊟ Variables					
BadValChk	If BadVal('PI Buffer Subsystem_Queue> Capacity') The				
ServiceUp	'Process(pibufss)_% Processor Time' <> "I/O Timeout"				
⊟ Start triggers					
LowQueue_Capacity	ServiceUp And 'PI Buffer Subsystem_Queue> Capacity'	10 minutes	Critical		

```
ServiceUp And 'PI Buffer Subsystem_Queue> Capacity' < 'PI Buffer Subsystem_Queue> Capacity|Lo'  // Low disk space
alert
```

1. It is comparing the value of the **PI Buffer Subsystem Queue> Capacity** attribute to the **PI Buffer Subsystem Queue> Capacity|Lo** child attribute.
2. If the attribute value is less than the **Lo** threshold for 10 minutes, this event frame will be triggered. Note that is is scheduled periodic and runs every 1 minute.
3. **NOTE:** if you want to speed things up for the course, you can go into the **Low Queue Capacity** analysis on the **PI Buffer Subsystem** template and change the **True for** on the **LowQueue_Capacity** trigger to a lower number (1 or 2 min), and check in your changes. It will automatically update on the element.

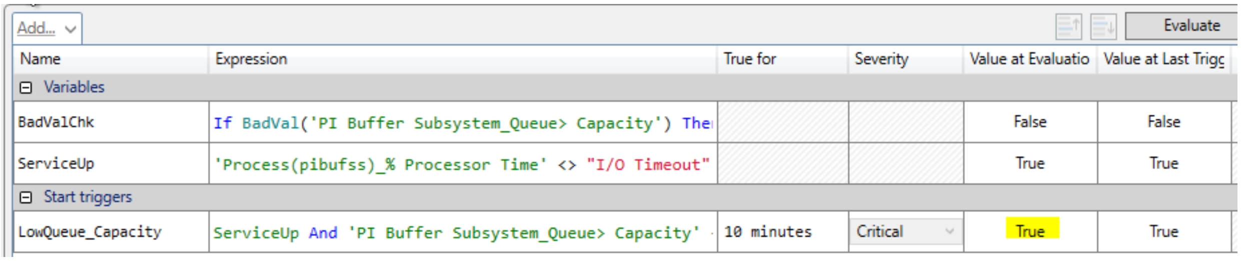
b. Select the **Attributes** tab in the viewer, and find the **PI Buffer Subsystem Queue> Capacity** attribute (it is under the Queues category). Expand the attribute to see the **Lo** child attribute. The threshold value is set to 7 days.



Category: Queues	
PI Buffer Subsystem_Queue> Capacity	765.74 d
Lo	7 d
Maximum	1E+05 d
Minimum	0 d

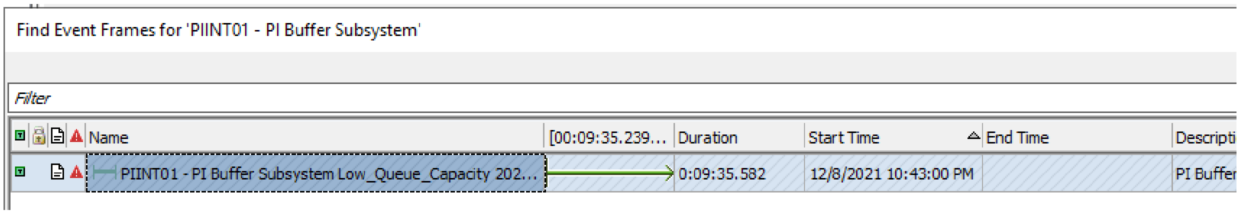
i. Remember that the triggering condition is when the **Capacity** is lower than the **Lo** threshold. Set **Lo** to **800 d**. Check in your changes.

c. Select the **Analyses** tab in the viewer and select the **Low Queue Capacity** event frame analysis. Click **Evaluate** and see that the value for the **LowQueue_Capacity** trigger is **True**. Note that it has to stay true for 10 mins for the event to trigger.



Name	Expression	True for	Severity	Value at Evaluatio	Value at Last Trigg
⊟ Variables					
BadValChk	If BadVal('PI Buffer Subsystem_Queue> Capacity') The			False	False
ServiceUp	'Process(pibufss)_% Processor Time' <> "I/O Timeout"			True	True
⊟ Start triggers					
LowQueue_Capacity	ServiceUp And 'PI Buffer Subsystem_Queue> Capacity'	10 minutes	Critical	True	True

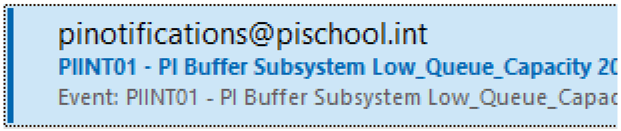
d. After the required time, select the **General** tab and click **Event Frames** – you will see a new event frame for **PI Buffer Subsystem Low_Queue_Capacity**. The blank **End Time** shows the event frame is still active.


Find Event Frames for 'PIINT01 - PI Buffer Subsystem'

Name	[00:09:35.239...	Duration	Start Time	End Time	Descripti
PIINT01 - PI Buffer Subsystem Low_Queue_Capacity 202...		0:09:35.582	12/8/2021 10:43:00 PM		PI Buffer

i. **Close** the **Event Frame** window.

e. Open Microsoft Outlook on the PIMONITOR01 machine. In the *student01@pischool.int* Inbox locate the email for the **PI Buffer Subsystem Low_Queue_Capacity** notification. Refresh if it is not showing.


pinotifications@pischool.int
PIINT01 - PI Buffer Subsystem Low_Queue_Capacity 20
Event: PIINT01 - PI Buffer Subsystem Low_Queue_Capac

i. Open the email and look at the content. The event name (which is also the email subject) shows which condition triggered the event. The table provides some key information on performance indicators. Below the table are instructions to troubleshoot the issue and a link to the event details view in PI Vision.

f. Select the **Attributes** tab in the viewer, and find the **PI Buffer Subsystem Queue> Capacity** attribute. Change the value for **Lo** back to **7**. Check in your changes.

i. Select the **Analyses** tab in the Viewer and select the **Low Queue Capacity** event frame analysis. Click **Evaluate** and see that the value for the **LowQueue_Capacity** trigger is now **False**.

ii. Select the **General** tab and click **Event Frames**. Confirm the event frame has closed – it will have an End Time now. Note the analysis is scheduled to run periodically every 1 minute so it may take up to a minute for the event frame to close.

iii. You should also get a **Notification is closed** email.

10. Other Answer Components

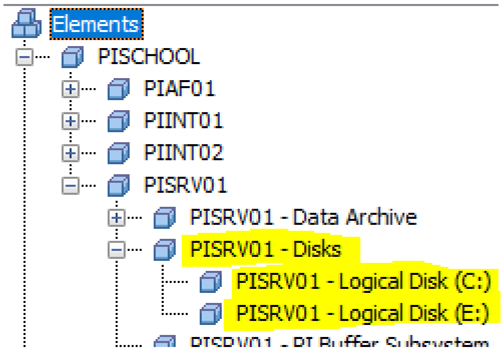
Each of the below components are included in the answer database. Look at the template for each component, and see how the element is configured in the hierarchy.

10.1. Server – disk monitoring

The **PSM Logical Disk** template can be used to monitor a logical disk connected to a Windows server. Analyses monitor and notify on the following excursions:

- High disk utilization
- Low disk space

The **PSM Logical Disk** element is created as a child of a **PSM Server (*)** element. If you have multiple disks you wish to monitor on a server you can create multiple Logical Disk elements. You may wish to use the **PSM Group (Components)** template to organize multiple disks under a server (as shown in the answer database below).



```
Elements
  PISCHOOL
    PIAF01
    PIINT01
    PIINT02
    PISRV01
      PISRV01 - Data Archive
      PISRV01 - Disks
        PISRV01 - Logical Disk (C:)
        PISRV01 - Logical Disk (E:)
      PISRV01 - PI Buffer Subsystem
```

Several configuration items are inherited, and the following attributes require user input:

- Disk Name

10.2. Server – Windows service

The **PSM Service** template can be used to monitor the availability of any Windows service that registers a performance counter for processor time usage. Analyses monitor and notify on the following excursions:

- Service not running

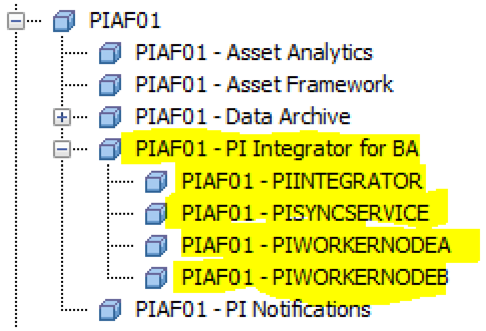
The **PSM Service** template is created as a child of a **PSM Server (*)** element. If you have multiple services you wish to monitor on a server you can create multiple Service elements. You may wish to use the **PSM Group (Components)** template to organize multiple services under a server.

Several configuration items are inherited, and the following attributes require user input:

- Process Name
- Service Display Name
- Service Name

In the course answer database, the **PSM Service** template is used with the PI Integrator for BA. PI Integrators do not expose any PI System-specific Windows performance counters for monitoring, so the ability to monitor them with the PSM example kit is limited. You can use the **PSM Service** template to monitor the availability of the Windows services for the PI Integrator.

Use the **PSM Group (Component)** organizational template to create a grouping for the various PI Integrator services, and name it accordingly (**PI Integrator for BA** below). You would then create an element using the PSM Service template for each service under the group.



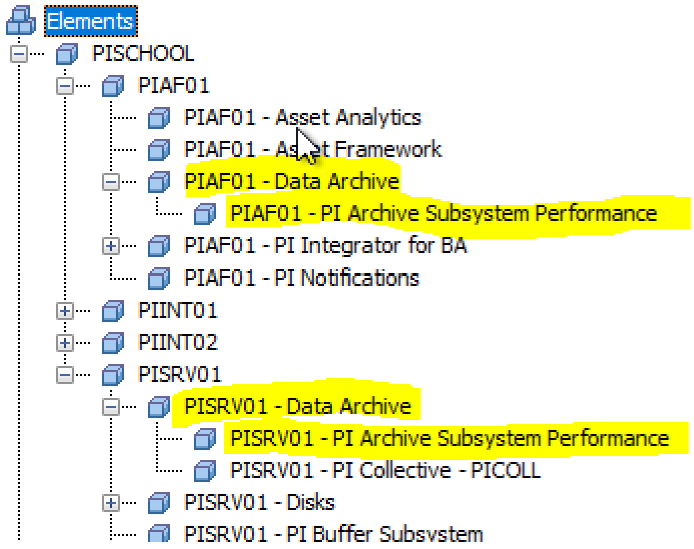
- PIAF01
  - PIAF01 - Asset Analytics
  - PIAF01 - Asset Framework
  - PIAF01 - Data Archive
  - PIAF01 - PI Integrator for BA
    - PIAF01 - PIINTEGRATOR
    - PIAF01 - PISYNCSERVICE
    - PIAF01 - PIWORKERNODEA
    - PIAF01 - PIWORKERNODEB
  - PIAF01 - PI Notifications

10.3. PI Archive subsystem

The **PSM PI Archive Subsystem Performance** template can be used to monitor the performance of the Archive subsystem on a Data Archive. Analyses monitor and notify on the following excursions:

- High ratio of archive read calls to disk instead of memory

The **PSM PI Archive Subsystem Performance** element is created as a child of a **PSM Data Archive** element. It can be used on a standalone Data Archive or any member of a collective.



- Elements
  - PISCHOOL
    - PIAF01
      - PIAF01 - Asset Analytics
      - PIAF01 - Asset Framework
      - PIAF01 - Data Archive
        - PIAF01 - PI Archive Subsystem Performance
      - PIAF01 - PI Integrator for BA
      - PIAF01 - PI Notifications
    - PIINT01
    - PIINT02
    - PISRV01
      - PISRV01 - Data Archive
        - PISRV01 - PI Archive Subsystem Performance
        - PISRV01 - PI Collective - PICOLL
      - PISRV01 - Disks
      - PISRV01 - PI Buffer Subsystem

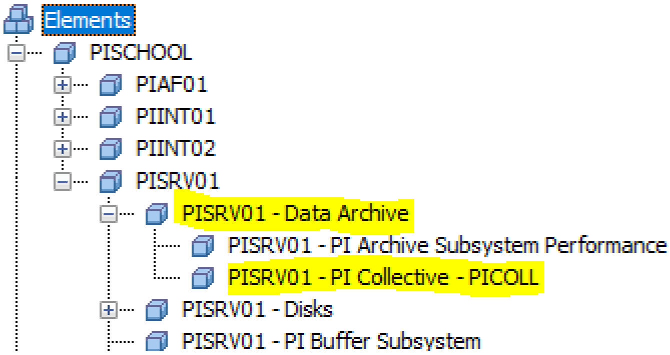
Several configuration items are inherited, but there are no attributes requiring user input.

10.4. Data Archive collective

The **PSM PI Collective** template can be used to monitor the functionality of a Data Archive collective. Analyses monitor and notify on the following excursions:

- PI collective not running normally

The **PSM PI Collective** element is created as a child of a **PSM Data Archive** element. It should only be created on the primary Data Archive in a collective.



- Elements
  - PISCHOOL
    - PIAF01
    - PIINT01
    - PIINT02
    - PISRV01
      - PISRV01 - Data Archive
        - PISRV01 - PI Archive Subsystem Performance
        - PISRV01 - PI Collective - PICOLL
      - PISRV01 - Disks
      - PISRV01 - PI Buffer Subsystem

Several configuration items are inherited, and the following attributes require user input:

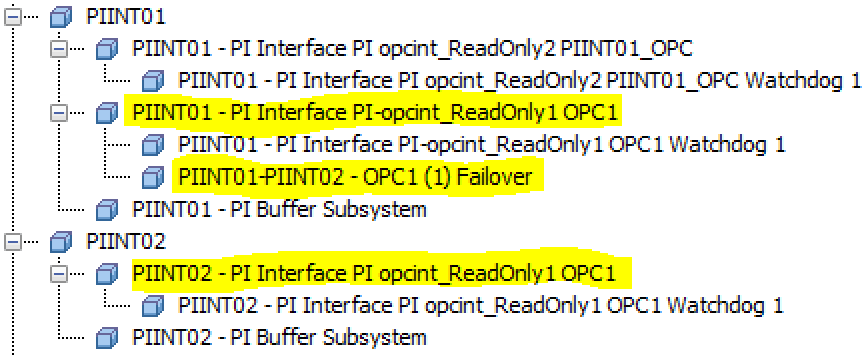
- Collective Name

10.5. Interface failover

The **PSM PI Interface Failover** template can be used to monitor the functionality of UniInt phase 2 failover (UFO) on an interface pair. Analyses monitor and notify on the following excursions:

- Failover occurred
- Failover state problem

The **PSM PI Interface Failover** element is created as a child of a **PSM Interface** element. It should only be created on one member of a failover pair (usually the one with the lowest Failover ID). Monitoring for interface failover is achieved by reading the values for the UFO failover state PI Tags from the Data Archive the interface is writing data to.

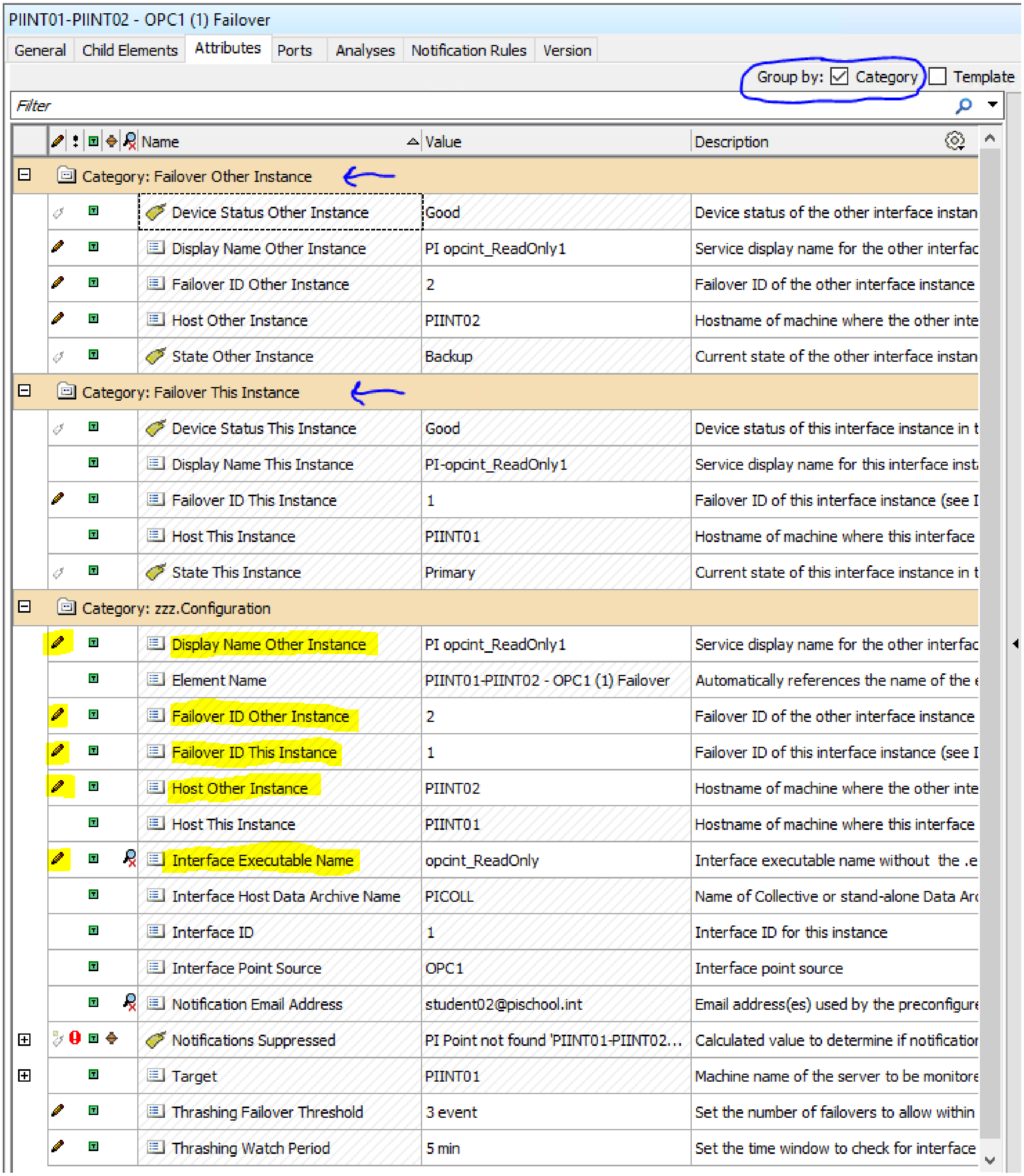


- PIINT01
  - PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
    - PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Watchdog 1
  - PIINT01 - PI Interface PI-opcint_ReadOnly1 OPC1
    - PIINT01 - PI Interface PI-opcint_ReadOnly1 OPC1 Watchdog 1
    - PIINT01-PIINT02 - OPC1 (1) Failover
  - PIINT01 - PI Buffer Subsystem
- PIINT02
  - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1
    - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1 Watchdog 1
  - PIINT02 - PI Buffer Subsystem

Several configuration items are inherited, and there are several attributes that require user input as noted in their default values. Only input manually on the ones with the configuration (pencil) icon (highlighted in yellow below).

- Display Name Other Instance
- Failover ID Other Instance
- Failover ID This Instance
- Host Other Instance
- Interface Executable Name

Note that the **PSM PI Interface Failover** template is the only PSM template to apply more than one category to an attribute. This means an attribute may show multiple times in the **Attribute** tab if it is shown with **Group by Category**.



PIINT01-PIINT02 - OPC1 (1) Failover

Group by: ☑ Category ☐ Template

Filter

Name	Value	Description
Category: Failover Other Instance		
Device Status Other Instance	Good	Device status of the other interface instan
Display Name Other Instance	PI opcint_ReadOnly1	Service display name for the other interfac
Failover ID Other Instance	2	Failover ID of the other interface instance
Host Other Instance	PIINT02	Hostname of machine where the other inte
State Other Instance	Backup	Current state of the other interface instan
Category: Failover This Instance		
Device Status This Instance	Good	Device status of this interface instance in t
Display Name This Instance	PI-opcint_ReadOnly1	Service display name for this interface inst
Failover ID This Instance	1	Failover ID of this interface instance (see I
Host This Instance	PIINT01	Hostname of machine where this interface
State This Instance	Primary	Current state of this interface instance in t
Category: zzz.Configuration		
Display Name Other Instance	PI opcint_ReadOnly1	Service display name for the other interfac
Element Name	PIINT01-PIINT02 - OPC1 (1) Failover	Automatically references the name of the e
Failover ID Other Instance	2	Failover ID of the other interface instance
Failover ID This Instance	1	Failover ID of this interface instance (see I
Host Other Instance	PIINT02	Hostname of machine where the other inte
Host This Instance	PIINT01	Hostname of machine where this interface
Interface Executable Name	opcint_ReadOnly	Interface executable name without the .e
Interface Host Data Archive Name	PICOLL	Name of Collective or stand-alone Data Arc
Interface ID	1	Interface ID for this instance
Interface Point Source	OPC1	Interface point source
Notification Email Address	student02@pischool.int	Email address(es) used by the preconfigure
Notifications Suppressed	PI Point not found 'PIINT01-PIINT02…	Calculated value to determine if notification
Target	PIINT01	Machine name of the server to be monitore
Thrashing Failover Threshold	3 event	Set the number of failovers to allow within
Thrashing Watch Period	5 min	Set the time window to check for interface

This was done in order to both keep the configuration items together in the **zzz.Cofiguration** category, while also visually grouping the multiple attributes for each failover instance in its own category (**Failover Other**

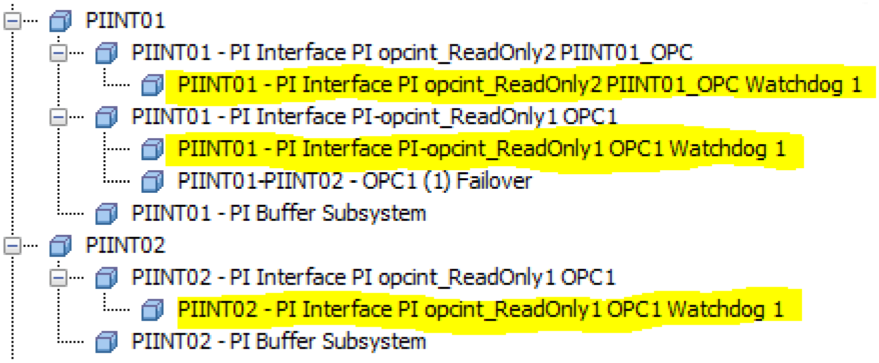
Instance, Failover This Instance). When you are manually entering configuration values for any of these attributes, you only have to enter it in one place (doesn't matter which one) and it will show in both.

10.6. Interface watchdog

The **PSM Watchdog Tag for PI Interfaces** template can be used to capture data source issues not covered by the interface's Device Status = Device(s) in error. For example, the interface is able to connect to the data source, but the data source is not sending any updates, or devices upstream of the interface's data source are having communications issues. Analyses monitor and notify on the following excursions:

- Stale or flatline data

The **PSM Watchdog Tag for PI Interfaces** element is created as a child of a **PSM Interface** element. It can be used on standalone or failover interfaces. When used on a failover pair, it should be created on both members. You can create as many watchdog child elements as you like for monitoring.



- PIINT01
  - PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC
    - PIINT01 - PI Interface PI opcint_ReadOnly2 PIINT01_OPC Watchdog 1
  - PIINT01 - PI Interface PI-opcint_ReadOnly1 OPC1
    - PIINT01 - PI Interface PI-opcint_ReadOnly1 OPC1 Watchdog 1
    - PIINT01-PIINT02 - OPC1 (1) Failover
  - PIINT01 - PI Buffer Subsystem
- PIINT02
  - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1
    - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1 Watchdog 1
  - PIINT02 - PI Buffer Subsystem

Several configuration items are inherited, and the following attribute requires user input:

- Tag Name (child attribute of Watchdog Tag)

The watchdog tag used for configuration should be different under each member of a failover pair. Since both members write to the same PI Tags (only one is ever actively writing), you can monitor watchdogs from either or both interface instance elements.

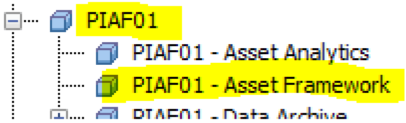
Using different tags on each failover member gives you full monitoring of all watchdogs as long as both interface instances are running properly. Due to notifications suppression, if one server or instance of a failover pair goes down and stays down, it will suppress all monitoring on the child watchdog tags so you may lose half your monitoring. It is recommended to set up multiple watchdogs on each instance.

10.7. Asset Framework (AF)

The **PSM Asset Framework** template can be used to monitor general health of the Asset Framework system, including the backend SQL Server. Analyses monitor and notify on the following excursions:

- Asset Framework general health

The **PSM Asset Framework** element is created as a child of a **PSM Server (*)** element.



PIAF01
PIAF01 - Asset Analytics
PIAF01 - Asset Framework
PIAF01 - Data Archive

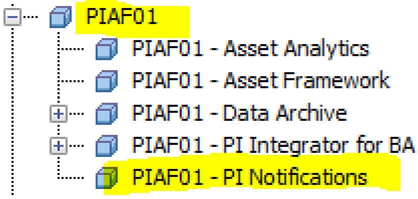
Several configuration items are inherited, but there are no attributes requiring user input.

10.8. PI Notifications

The **PSM PI Notifications** template can be used to monitor functionality of the PI Notifications service. Analyses monitor and notify on the following excursions:

- High number of failed sends in past 5 mins
- High number of rules in error in past 5 mins

The **PSM PI Notifications** element is created as a child of a **PSM Server (*)** element.



PIAF01
PIAF01 - Asset Analytics
PIAF01 - Asset Framework
PIAF01 - Data Archive
PIAF01 - PI Integrator for BA
PIAF01 - PI Notifications

Several configuration items are inherited, but there are no attributes requiring user input.

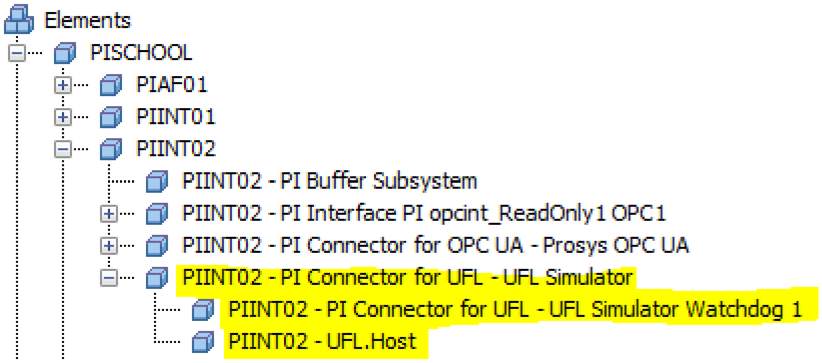
10.9. PI Integrator for Business Analytics

See *Server – Windows service*

10.10. PI Connectors

The PI Connectors do not expose any custom Windows performance counters for monitoring, so the ability to monitor them with the PSM example kit is limited. However, you can use the **PSM Service** template to monitor the availability of the Windows service for a PI Connector, and you can use the **PSM Watchdog Tag for PI Connectors** to monitor the availability of the data source.

To accommodate this, instead of using the **PSM Group (Component)** organizational template to create a grouping for the PI Connector service, use the **PSM PI Connector** template. The **PSM PI Connector** element is created as a child of a **PSM Server (*)** element. The **PSM Service** and **PSM Watchdog Tag for PI Connectors** elements are created as children of a **PSM PI Connector** element.



- Elements
  - PISCHOOL
    - PIAF01
    - PIINT01
    - PIINT02
      - PIINT02 - PI Buffer Subsystem
      - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1
      - PIINT02 - PI Connector for OPC UA - Prosys OPC UA
      - PIINT02 - PI Connector for UFL - UFL Simulator
        - PIINT02 - PI Connector for UFL - UFL Simulator Watchdog 1
        - PIINT02 - UFL.Host

Several configuration items are inherited, and the following attributes require user input:

- Connector Host Data Archive Name
- Connector Product Name
- Data Source Name

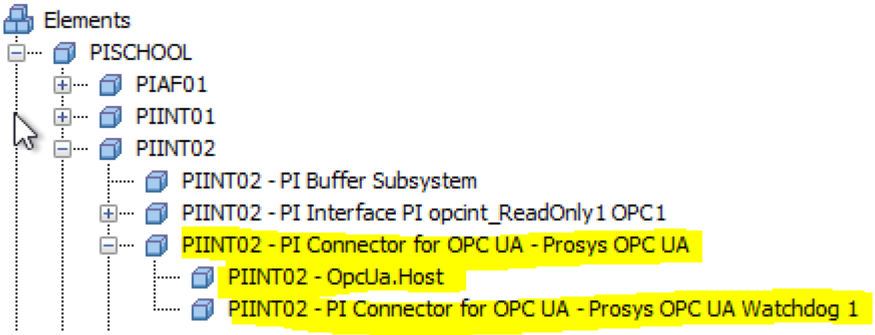
For more information on the **PSM Service** template and usage see *Server – Windows service*. The **PSM Watchdog Tag for PI Connectors** template is very similar to the **PSM Watchdog Tag for PI Interfaces** template, but with fewer configuration items. See *Interface watchdog* for more information.

10.10.1. PI Connector for OPC UA

The PI Connector for OPC UA is a special case – it actually creates a few PI Tags for server state and times on the Data Archive it is writing data to. Additional monitoring of the PI Connector for OPC UA is achieved by reading the values for those PI Tags from the host Data Archive. The **PSM Service** template can be added to monitor the availability of the Windows service for a PI Connector, and you can use the **PSM Watchdog Tag for PI Connectors** to monitor the availability of the data source.
Analyses monitor and notify on the following excursions:

- High number of failed sends in past 5 mins

The **PSM PI Connector for OPC UA** element is created as a child of a **PSM Server (*)** element. This template is derived from the **PSM PI Connector** template and adds the attributes required to reference the PI Connector host Data Archive. The **PSM Service** and **PSM Watchdog Tag for PI Connectors** elements are then created as children of a **PSM PI Connector for OPC UA** element.



- Elements
  - PISCHOOL
    - PIAF01
    - PIINT01
    - PIINT02
      - PIINT02 - PI Buffer Subsystem
      - PIINT02 - PI Interface PI opcint_ReadOnly1 OPC1
      - PIINT02 - PI Connector for OPC UA - Prosys OPC UA
        - PIINT02 - OpcUa.Host
        - PIINT02 - PI Connector for OPC UA - Prosys OPC UA Watchdog 1

Several configuration items are inherited or have default values, and the following attributes require user input:

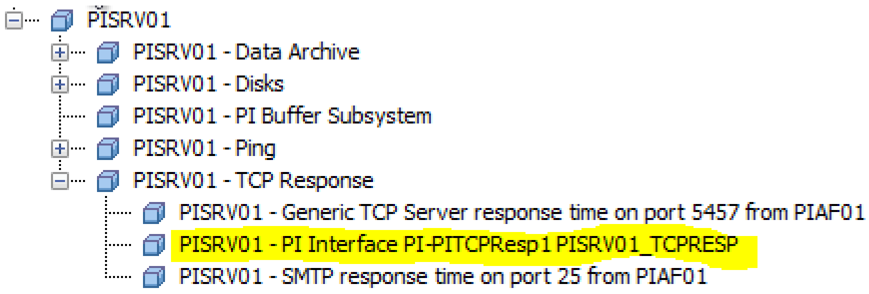
- Connector Host Data Archive Name

- ○ Data Source Name

10.11. Other PI System diagnostic interfaces

If you decide to deploy the PI Interface for PING or PI Interface for TCP Response, there are templates in the PSM example kit that can be used with these interfaces to provide monitoring. These templates are different than all the other PSM templates as they are not used to monitor the component they are for – i.e., they do not monitor the availability of the PI Interface for PING or TCP Response, and do not use the PIPerfMon interface to gather data. Instead they monitor whatever that interface gathers data on (ping or TCP response times).

Once you have deployed and configured the PI Interface for PING or TCP Response, you can deploy an element for the associated PSM template and configure it appropriately. If you wish to monitor the PI Interface for PING or TCP Response itself (to be alerted if it has failures or issues), you can use the PSM PI Interface template, and configure as needed.



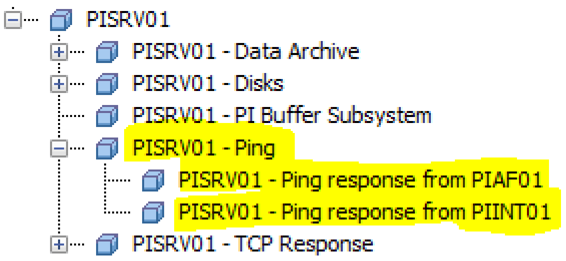
```
PISRV01
    PISRV01 - Data Archive
    PISRV01 - Disks
    PISRV01 - PI Buffer Subsystem
    PISRV01 - Ping
    PISRV01 - TCP Response
        PISRV01 - Generic TCP Server response time on port 5457 from PIAF01
        PISRV01 - PI Interface PI-PITCPResp1 PISRV01_TCPRESP
        PISRV01 - SMTP response time on port 25 from PIAF01
```

10.11.1. PI Interface for PING

The **PSM Ping Response** template can be used to monitor the response time when one machine pings another. It is recommended to create the element for this template on the originator of the ping. Analyses monitor and notify on the following excursions:

- Slow or no reply to ping

The **PSM Ping Response** element is created as a child of a **PSM Server (*)** element. If you have multiple destinations you wish to monitor ping time for from this originator, you can create multiple **PSM Ping Response** elements. You may wish to use the **PSM Group (Components)** template to organize multiple pings under a server (as shown in the answer database below).



```
PISRV01
    PISRV01 - Data Archive
    PISRV01 - Disks
    PISRV01 - PI Buffer Subsystem
    PISRV01 - Ping
        PISRV01 - Ping response from PIAF01
        PISRV01 - Ping response from PIINT01
    PISRV01 - TCP Response
```

This template uses the configuration items in a slightly different way that the rest of the PSM templates. The following attributes require user input:

- Destination – destination of the ping

- Target – the originator of the ping – likely the Server this element is under
- Location 1 & 4 – these have default values, but may need to be adjusted to match configuration of the PI Interface for PING
- Ping Point Source – the point source used by the PI Interface for PING

For more details on the information required for these attributes consult the installation or user documentation for the PI Interface for PING.

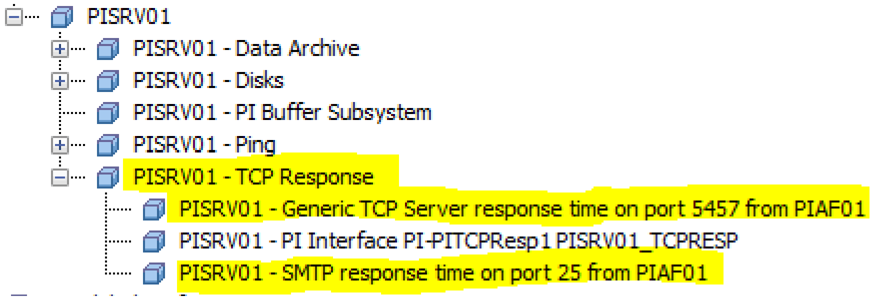
Note that the Target and Destination are used in the element naming pattern - once those have been entered check in changes and refresh.

10.11.2. PI Interface for TCP Response

The **PSM TCP Response** template can be used to monitor the response time for a specific server role and TCP/IP service port. It is recommended to create the element for this template on the originator of the request. Analyses monitor and notify on the following excursions:

- Slow or no reply to TCP request

The **PSM TCP Response** element is created as a child of a **PSM Server (*)** element. If you have multiple destinations you wish to monitor TCP response time for from this originator, you can create multiple **PSM TCP Response** elements. You may wish to use the **PSM Group (Components)** template to organize multiple TCP response elements under a server (as shown in the answer database below).



PISRV01
- PISRV01 - Data Archive
- PISRV01 - Disks
- PISRV01 - PI Buffer Subsystem
- PISRV01 - Ping
- PISRV01 - TCP Response
  - PISRV01 - Generic TCP Server response time on port 5457 from PIAF01
  - PISRV01 - PI Interface PI-PITCPResp1 PISRV01_TCPRESP
  - PISRV01 - SMTP response time on port 25 from PIAF01

This template uses the configuration items in a slightly different way that the rest of the PSM templates. The following attributes require user input:

- Destination – machine the request is being sent to
- Operation Type – operation the interface will perform
- Target – the originator of the request – likely the Server this element is under
- Location 1 - 4 – these have default values, but may need to be adjusted to match configuration of the PI Interface for PING
  - Location 2 is dependent on Operation Type, and will automatically be updated after the Operation Type attribute value is set and checked in
- Port – port to test
- TCP Response Point Source – the point source used by the PI Interface for TCP Response

For more details on the information required for these attributes consult the installation or user documentation for the PI Interface for TCP Response.

Note that the Target, Operation Type, Port, and Destination are used in the element naming pattern - once those have been entered check in changes and refresh.

10.12. Notification suppression

Notification suppression is functionality that detects whether event frames should be created and notifications sent or not when an analysis runs. It helps prevent unnecessary event frames and nuisance notifications when an entire server or component goes down, or if a user wants to turn off this functionality for a specific component.

Notifications suppression is turned on when either of the following are true:

- Automatically if the overall component is detected to be down (ie: a server, service, etc. crashes, fails, or is stopped)
- Manually if a user flips the flag to turn on suppression (e.g., user chooses to suppress notifications)

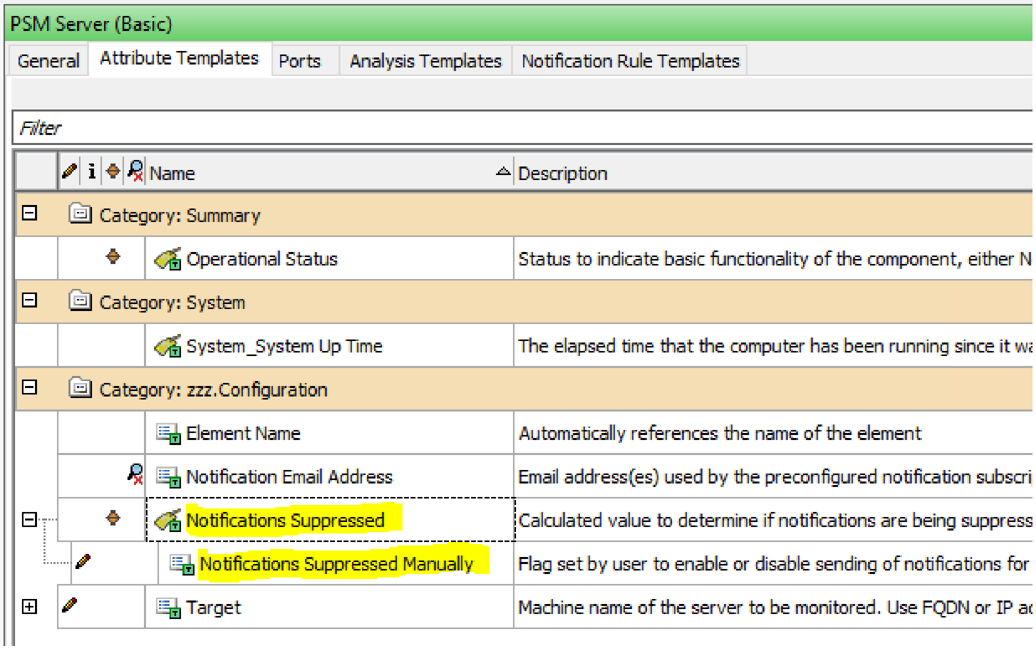
Notifications suppression is automatically turned off when both of the following are true:

- The server is detected as running
- Suppression is not manually turned on

Currently the functionality only applies to servers, interface instances, interface failover, and interface watchdogs.

- If a server fails or is stopped, one event frame will be created and one notification will be sent out to advise the server is down. No other event frames or notifications will be created for other server-level failures.
  - If there are interface instance components on that server, they will not create any event frames or notifications for failure (this includes any of their failover or watchdog child components).
  - The event frame and notification will close where the criteria for turning off notification suppression (listed above) is true.
  - All other components will continue to be monitored, create event frames, and send notifications based on their specific analyses.
- If an interface instance fails or is stopped, but the server is still up and not suppressed, one event frame will be created and one notification will be sent out to advise the interface is down. No other event frames or notifications will be created for other interface-level failures.
  - If there are failover or watchdog child components on that interface, they will not create any event frames or notifications for failure.
  - The event frame and notification will close where the criteria for turning off notification suppression (listed above) is true.
  - All other interface instances (and their child failover or watchdog components) on the same machine will continue to be fully monitored.
- If the manual suppression flag is turned on (equals **True**) on a server or interface prior to it being stopped, no event frames or notifications will be created for it or any of its child components.
  - Monitoring for this component and its children will resume when the criteria for turning off notification suppression (listed above) is true.

The PSM Server (*) template has 2 attributes related to notifications suppression:

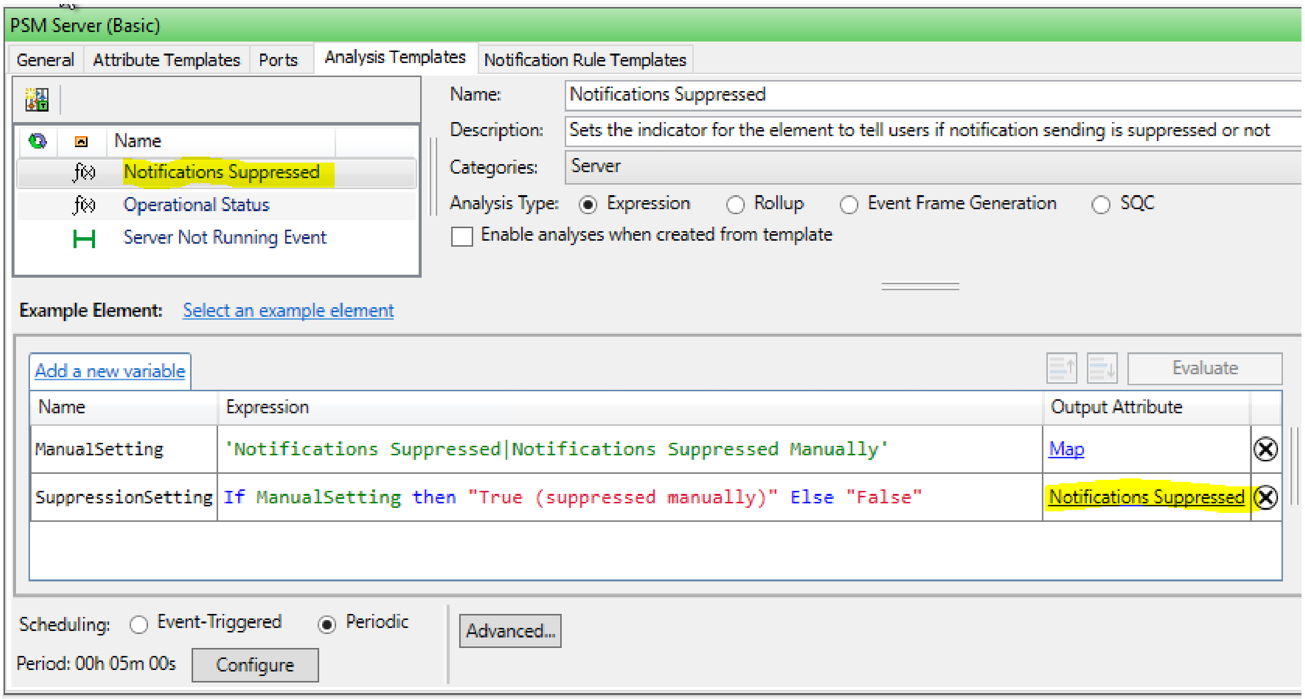


Name	Description
Category: Summary	
Operational Status	Status to indicate basic functionality of the component, either N
Category: System	
System_System Up Time	The elapsed time that the computer has been running since it wa
Category: zzz.Configuration	
Element Name	Automatically references the name of the element
Notification Email Address	Email address(es) used by the preconfigured notification subscri
Notifications Suppressed	Calculated value to determine if notifications are being suppress
Notifications Suppressed Manually	Flag set by user to enable or disable sending of notifications for
Target	Machine name of the server to be monitored. Use FQDN or IP ac

a. **Notifications Suppressed** – analysis output indicating whether notifications are turned off for this object either manually or automatically

b. **Notifications Suppressed Manually** – attribute to allow users to manually turn off notifications for this object and its children, to allow for planned server outages. This has a default value of **False** on every template.

The **Notifications Suppressed** expression analysis determines the value for the **Notifications Suppressed** attribute above.



Name: Notifications Suppressed

Description: Sets the indicator for the element to tell users if notification sending is suppressed or not

Categories: Server

Analysis Type: ● Expression ○ Rollup ○ Event Frame Generation ○ SQC

□ Enable analyses when created from template

Example Element: Select an example element

Name	Expression	Output Attribute
ManualSetting	'Notifications Suppressed|Notifications Suppressed Manually'	Map
SuppressionSetting	If ManualSetting then "True (suppressed manually)" Else "False"	Notifications Suppressed

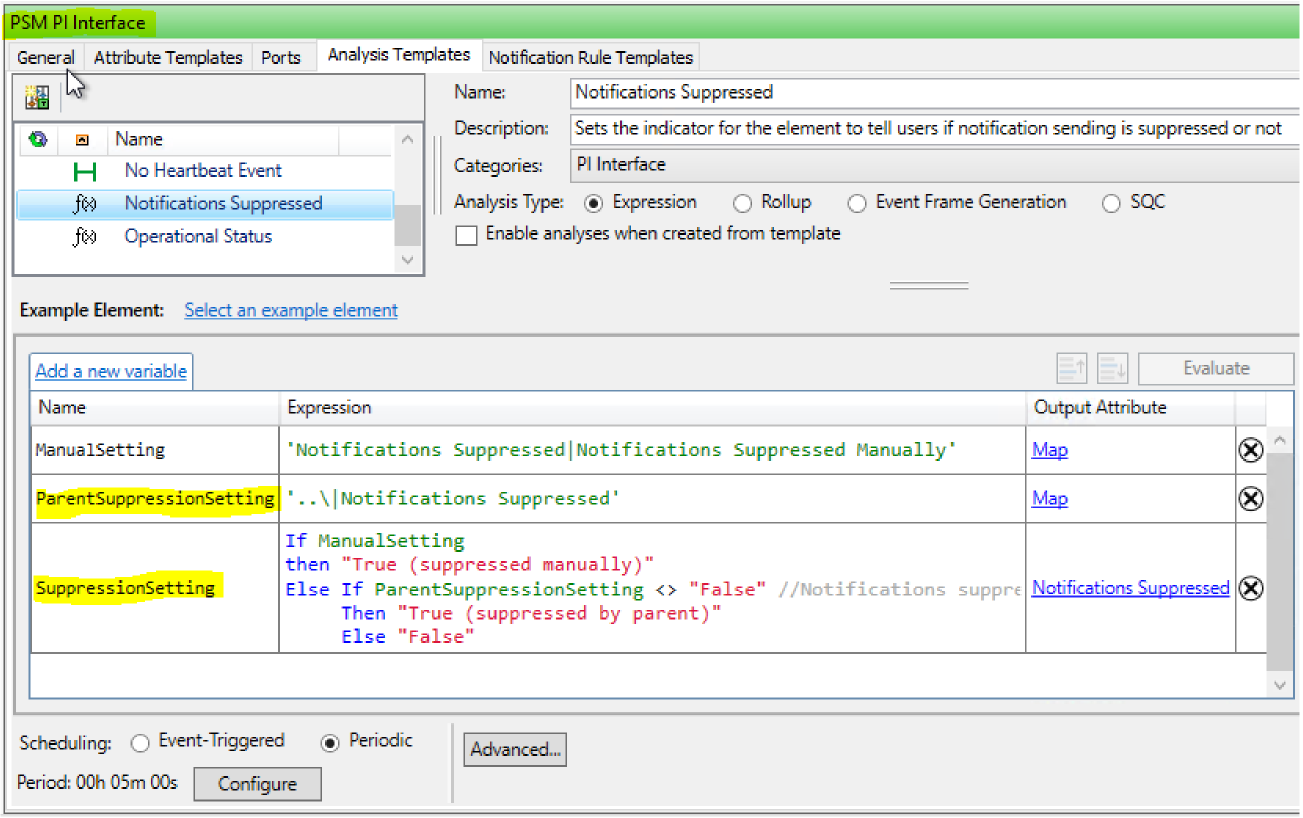
Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 05m 00s

a. Variable **ManualSetting** gets the value of the **Notifications Suppressed Manually** attribute. This attribute is set to **False** by default, and users can flip it to **True** if they wish to manually turn off notifications for this element and all of its interface children.

b. Variable **SuppressionSetting** sets the overall value for the **Notifications Suppressed** attribute. In this template the only thing that contributes to the overall **Notifications Suppressed** value is its own manual suppression flag.

Applicable child components include the same notification suppression attributes, but the analysis logic includes additional factors.



Name: Notifications Suppressed

Description: Sets the indicator for the element to tell users if notification sending is suppressed or not

Categories: PI Interface

Analysis Type: ● Expression ○ Rollup ○ Event Frame Generation ○ SQC

□ Enable analyses when created from template

Example Element: Select an example element

Name	Expression	Output Attribute
ManualSetting	'Notifications Suppressed|Notifications Suppressed Manually'	Map
ParentSuppressionSetting	'..\\|Notifications Suppressed'	Map
SuppressionSetting	If ManualSetting then "True (suppressed manually)" Else If ParentSuppressionSetting <> "False" //Notifications suppre Then "True (suppressed by parent)" Else "False"	Notifications Suppressed

Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 05m 00s

This same analysis on a child equipped with notifications suppression also includes:

a. Variable **ParentSuppressionSetting** gets the value of the **Notifications Suppressed** attribute for the parent element

b. Variable **SuppressionSetting** sets the overall value for the **Notifications Suppressed** attribute. In child templates, it differentiates between itself being manually suppressed vs. it being automatically suppressed due to its parent (whether the parent was manually or automatically suppressed doesn't matter).

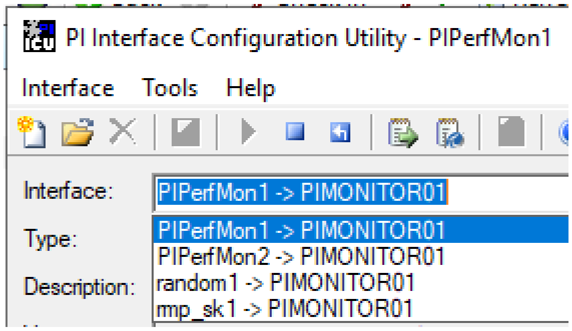
Note that this analysis on both parents and children is periodically scheduled to run every 5 minutes, so if a user turns on or off the flag for Notifications Suppressed Manually (at any level) it can take up to 5 minutes for that change to get picked up by the system.

For further information on the **Operational Status** analysis, see *Exercise 3: PI Interface – failed interface.*

11. Appendices

11.1. How to find the PIPerfMon point source

1. To get the PIPerfMon point source, log onto the machine where PIPerfMon is installed (PIMONITOR01 in the course). Open ICU (Interface Configuration Utility) and in the **Interface** box click the down arrow on the right and select **PIPerfMon1 -> PIMONITOR01**:



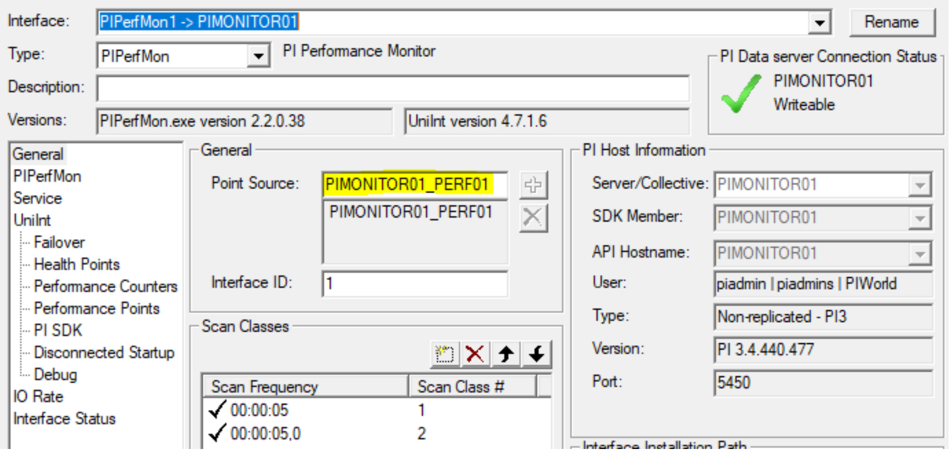
Interface: PIPerfMon1 -> PIMONITOR01

Type:

Description:

PIPerfMon1 -> PIMONITOR01
PIPerfMon2 -> PIMONITOR01
random1 -> PIMONITOR01
rmp_sk1 -> PIMONITOR01

2. On the **General** tab, note the **Point Source**:



Interface: PIPerfMon1 -> PIMONITOR01

Type: PIPerfMon PI Performance Monitor

Description:

Versions: PIPerfMon.exe version 2.2.0.38 UniInt version 4.7.1.6

PI Data server Connection Status: PIMONITOR01 Writeable

General

Point Source: PIMONITOR01_PERF01

PIMONITOR01_PERF01

Interface ID: 1

Scan Classes

Scan Frequency	Scan Class #
✓ 00:00:05	1
✓ 00:00:05,0	2

PI Host Information

Server/Collective: PIMONITOR01

SDK Member: PIMONITOR01

API Hostname: PIMONITOR01

User: piadmin | piadmins | PIWorld

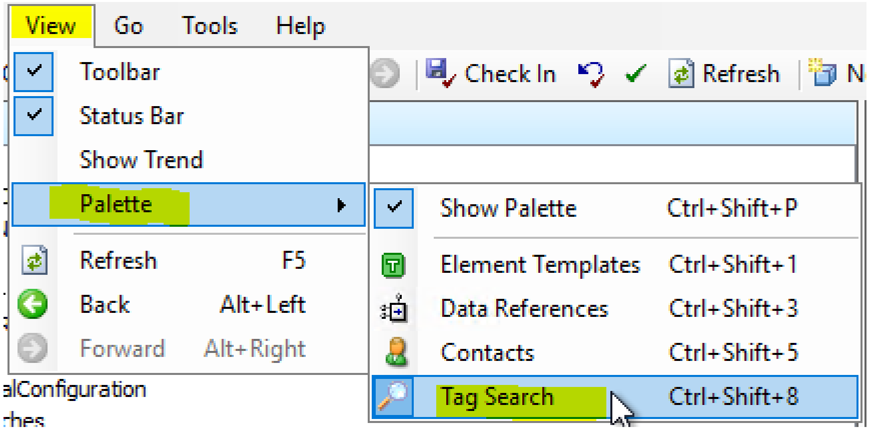
Type: Non-replicated - PI3

Version: PI 3.4.440.477

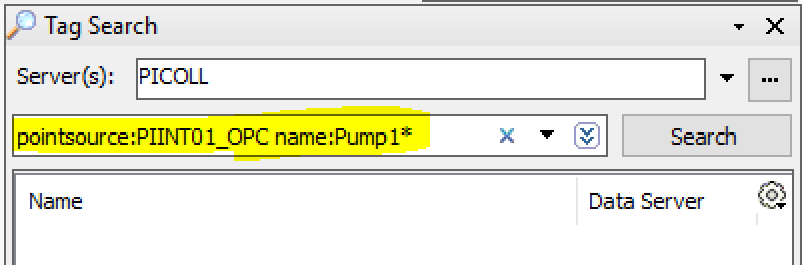
Port: 5450

11.2. How to find a good watchdog for interface data source monitoring

1. From the **View** menu at the top of PSE, select **Palette** > **Tag Search**.



2. The Tag Search palette will open on the right. Ensure **the proper** server is selected – this would be the Data Archive or collective containing the data source tag.
3. In the search box enter search criteria to find your watchdog PI tag. For example, in the course system use the following search criteria: **pointsource:PIINT01_OPC name:Pump1***
   Note: tag attributes will show in a popup as you start typing and you can select the matching item from a list. Each attribute should be separated by a space.

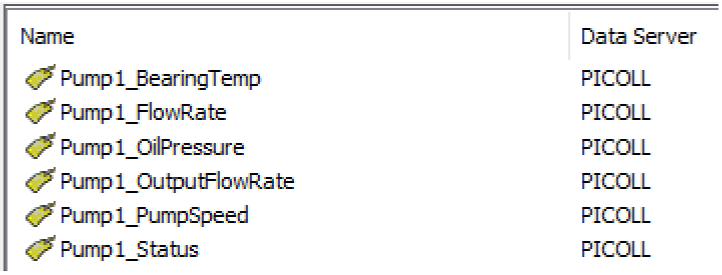


Tag Search

Server(s): PICOLL

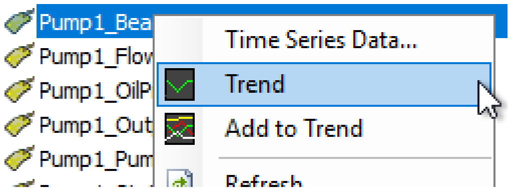
pointsource:PIINT01_OPC name:Pump1*

4. Click **Search** and the PI Tags matching the search criteria will show in the list.

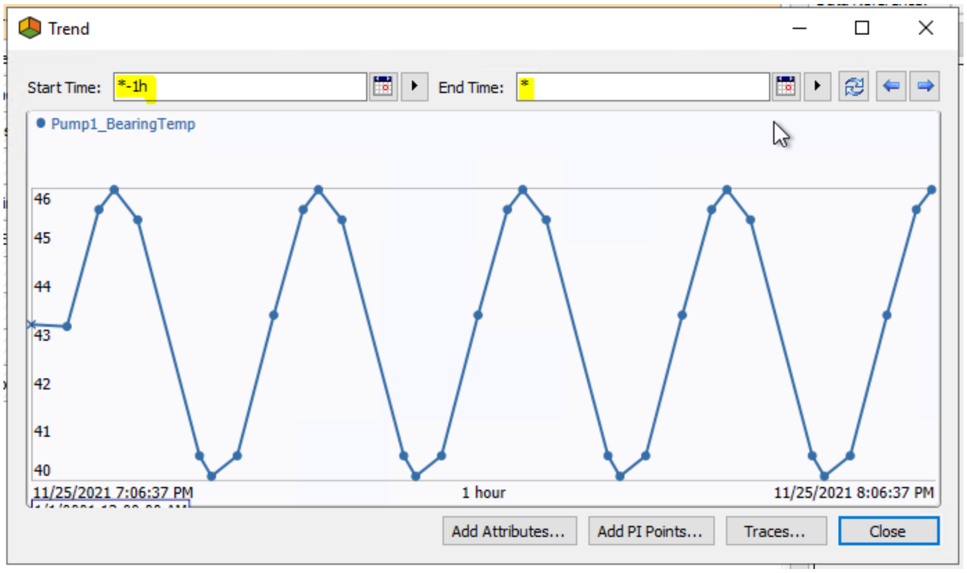


Name	Data Server
Pump1_BearingTemp	PICOLL
Pump1_FlowRate	PICOLL
Pump1_OilPressure	PICOLL
Pump1_OutputFlowRate	PICOLL
Pump1_PumpSpeed	PICOLL
Pump1_Status	PICOLL

5. Select **Pump1_BearingTemp**. **Right-click** and select **Trend**:



6. This will open a trend of the PI Tag value in a popup window. Change the start time to ‘***-1h**’ and the end time to ‘*****’ and hit **Enter** to see the values over the past hour.



Trend

Start Time: *-1h

End Time: *

7. Confirm from the trend that the PI Tag is appropriate as a watchdog – does it get new values regularly and does the value change frequently? These are important characteristics because you don’t want to

use a PI Tag that will set off a stale value alert because it doesn't change fast enough. Close the trend window.

8. It is also a good idea to turn compression off for a watchdog PI Tag, to ensure you get frequent updates. This may involve changing the PI Tag configuration of the watchdog tag itself.