2016 OSIsoft TechCon

Condition-Based Maintenance
with PI AF

OSIsoft, LLC
777 Davis St., Suite 250
San Leandro, CA 94577 USA
Tel: (01) 510-297-5800
Web: http://www.osisoft.com

Published: April 11, 2016

Condition-based Maintenance with PI AF – Hands-on Lab – OSIsoft TechCon 2016

Lead: Gopal GopalKrishnan, P.E., Solution Architect

Lead: Keith Pierce, Solution Architect

Instructor: LP Page-Morin, Sr. Systems Engineer

Instructor: François Pelletier-Bouchard, Systems Engineer

Table of Contents

Contents

Table of Contents 3
Summary 5
Exercise 1: Usage-based maintenance – Motor run hours 6
Exercise 2: Condition-based maintenance – Motor vibration high condition 7
Exercise 3: Predictive maintenance – Equipment failure, anomaly detection and RUL (remaining useful life) 9
The PI System Architecture for this Lab 10
Exercise 1 – Usage-Based Maintenance - Motor Run Hours 12
Step 1. Locate the Running Status and RunHours Attributes 12
Step 2. Examine the Run Hours Analysis 14
Step 3. Deploy the RunHours Analysis to the Whole Fleet 16
Step 4. Backfill the Newly Created Analyses 17
Step 5. Display the results in PI Coresight 18
Step 6. Report the Results in Microsoft Excel with PI DataLink (optional) 18
Step 7. Analyze the Results in Microsoft Power BI (optional) 19
Exercise 2 – Condition-Based Maintenance - Motor Vibration High Condition 20
Step 1. Examine the Available Information in PI AF 20
Step 2. Confirm the Event Frame Generation Analysis is Working 21
Step 3. Finalize the Configuration of the Notification Trigger Rule 22
Step 4. Customize the Student Contact Email Address 23
Step 5. Confirm your Subscription to the Email Notification 23
Step 6. Force the Triggering of the Notification Rule 24
Step 7. Check the Notification in PI Coresight 24
Step 8. Comment/Annotate the Notification in PI Coresight 26
Step 9. Confirm the Addition of the Comment to the Event Frame 26
Exercise 3 – Predictive Maintenance – Engine Failure 28

Summary

Condition-based maintenance (CBM) is a strategy where you monitor the actual condition of an asset to decide what maintenance needs to be done – see wiki for a broad definition. We also covered CBM at OSIsoft Users Conference 2015 – see Keeping Assets Healthy – PI System's Role in Asset Maintenance.

In this lab we show how to use sensor data to implement basic CBM tasks identified in the above talk:

- Exercise 1: Usage-based maintenance – Motor run hours
- Exercise 2: Condition-based maintenance – Motor vibration high condition
- Exercise 3: Predictive maintenance – Equipment failure, anomaly detection and remaining useful life

You have several critical motors in the plant. Currently, you use calendar-based schedule (weekly, monthly, quarterly, half-yearly, yearly etc.) to do preventive maintenance (PM) inspections and minor repairs/service for these motors.

The motors don't run continuously - their usage depends on the production schedule. As such, a motor may be idle for several hours during a day and sometimes even for several days or weeks. However, since maintenance personnel don't have visibility into their usage, motor PM tasks are performed even when they are not required – for example, weekly PM services on motors that have been idle for several weeks.

Login Step by Step:

1. Log into the machine PISRV01
2. Username: pischool\student01
3. Password: student

Exercise 1: Usage-based maintenance – Motor run hours

We use available sensor data for the motor – its *Running/Stopped* status collected by the PI System to create either desktop or web-based motor usage report - see sample screens below.

If a *Running/Stopped* status is not available, you can use other measurements such as motor amperage, rpm etc. to infer that a motor is running, for example, Motor amps>5 or Motor rpm>200 etc.

In Exercise 1, you will learn to create reports similar to those shown below:

Using PI ProcessBook and PI Coresight:

OSIsoft. **Report - Critical Motors - Run Hours** Last Update: 3-12-2016

Equipment	Daily Run Hours	Lifetime Run Hours	During Last Period	Period	Since Last Service	Last Service	Next Service
Agitator 1204	4.51	7,975	0	3mo	387	1/10/2016	11/10/2016
Agitator 1205	23.79	10,119	2,154	3mo	409	2/23/2016	10/3/2016
Agitator 1304	23.49	9,908	2,118	3mo	697	2/11/2016	12/13/2016
Agitator 1305	23.49	9,908	2,118	3mo	697	2/11/2016	12/1/2016
Fan 5163	19.71	8,554	1,174	3mo	2,664	10/1/2015	5/1/2016
Fan 5164	23.97	9,292	2,022	3mo	3,566	10/2/2015	5/2/2016
Fan 8144	14.44	9,839	2,112	3mo	3,635	10/5/2015	5/5/2016
Pump 3809	15.16	8,587	1,949	3mo	3,218	10/10/2015	5/10/2016
Pump 3810	23.97	9,618	2,079	3mo	3,837	9/23/2015	7/1/2016

Using Power BI or SQL Reporting:

REPORT - All Critical Motors - Run Hours

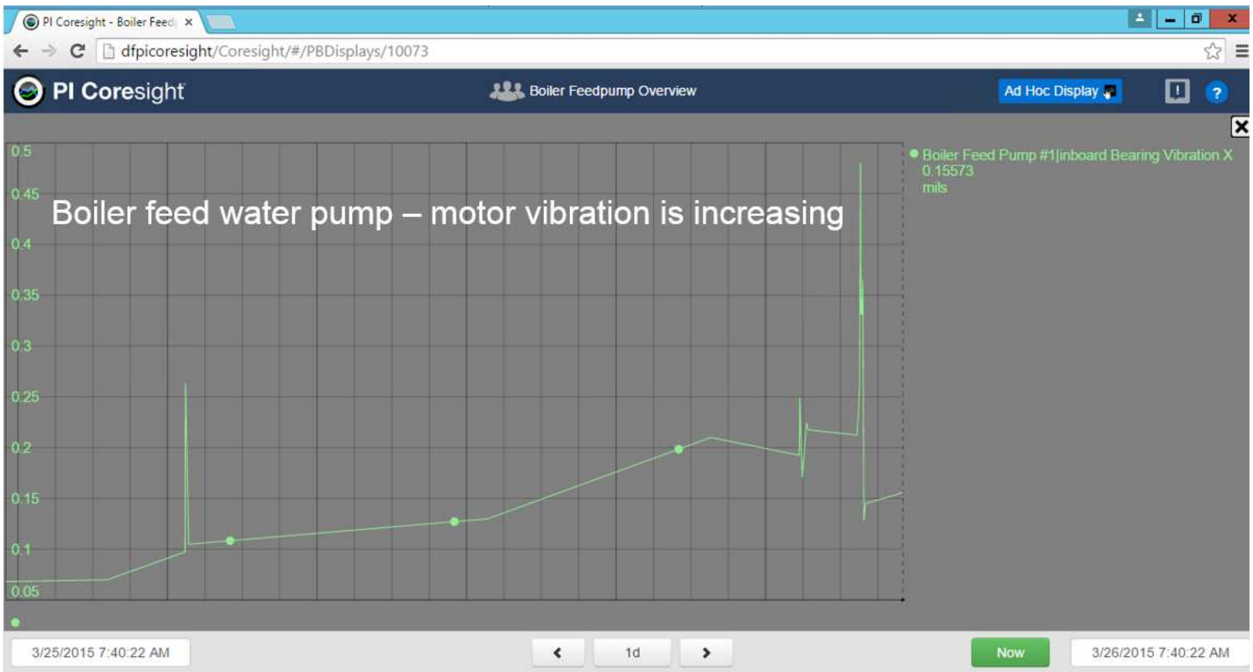
Equip...	Yesterday	Lifetime	Last Period	Period	Since Last Service	Last Service	Next Service
Fan 5164	24	7732	1153	90d	7732	10/2/2015	5/2/2016
Fan 5163	24	7874	1133	90d	1984	10/1/2015	5/1/2016

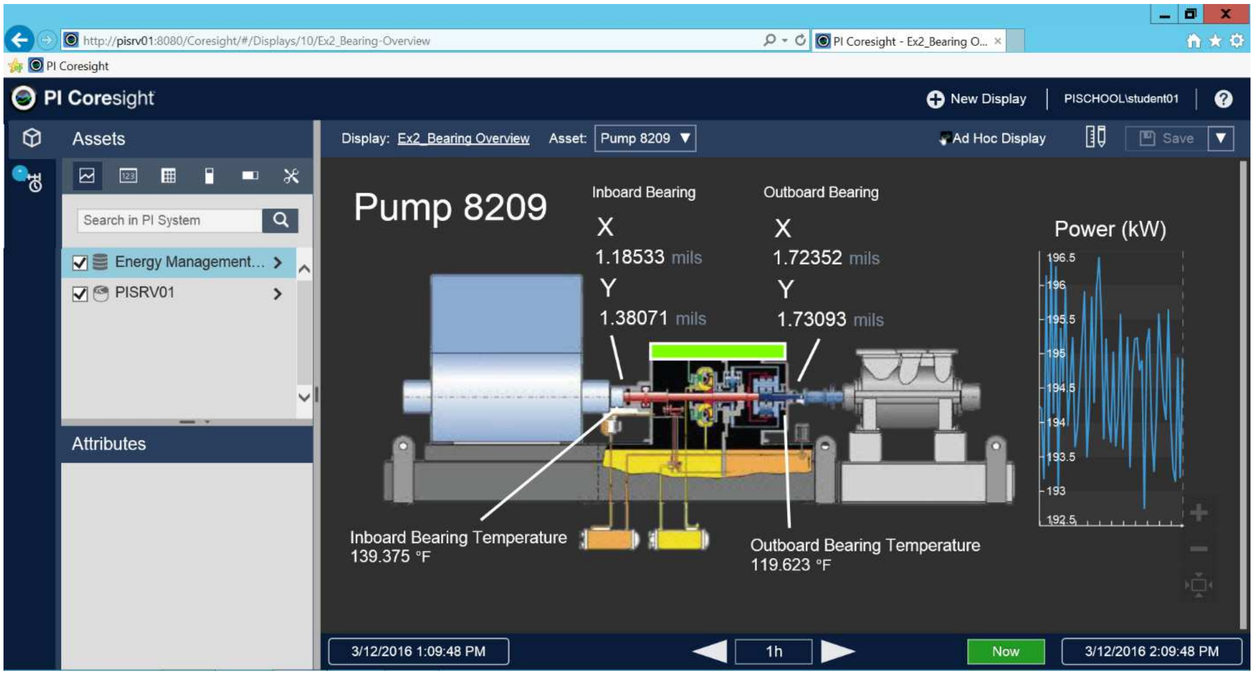
These reports provide easy access to equipment usage data for maintenance personnel and provide the basis to change over from calendar-based PM to usage-based PM and eliminate unnecessary and wasteful maintenance work.

Exercise 2: Condition-based maintenance – Motor vibration high condition

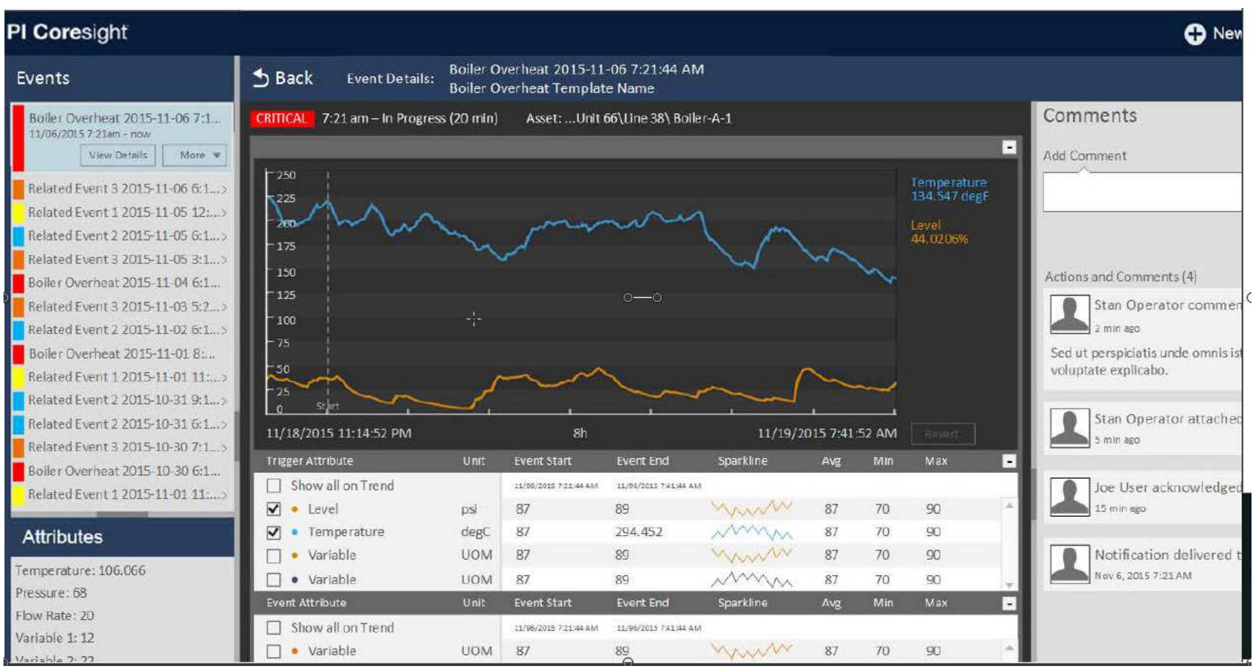
In condition-based maintenance, sensor data from the motors such as vibration, bearing temperature, windings temperature etc. are continuously monitored for warning signs that indicate imminent malfunction or failure. You are notified via email or sms regarding such conditions, say, motor vibrations that are higher than a threshold or rising rapidly.

The notification can include links to PI Coresight displays which can provide additional context to the triggering event, i.e. high vibration.   You can also acknowledge the event, enter comments and add appropriate media (picture, audio, video etc.) to the event – as shown by the screens below.





You can also have a watch-list with condition monitoring alerts from various equipment in PI Coresight 2016 – see below:

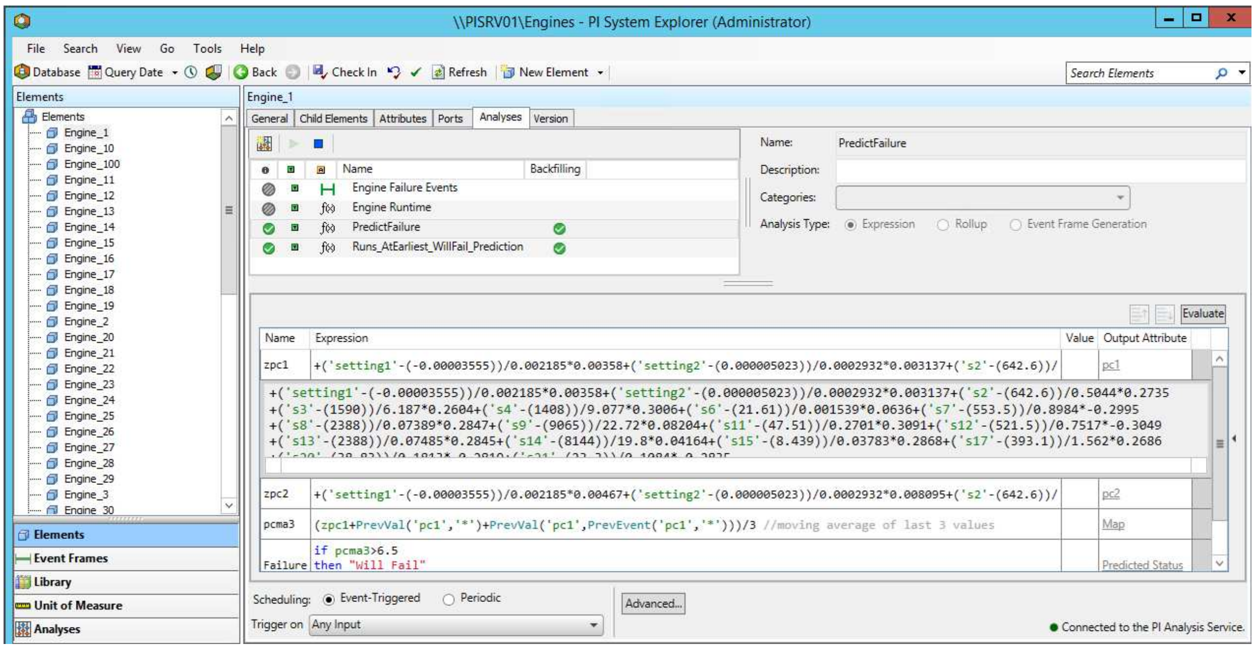


Exercise 3: Predictive maintenance – Equipment failure, anomaly detection and RUL (remaining useful life)

In this exercise, we review the predictive model created in *Data Science for Machine Learning and Predictions based on PI System data* lab (also referred as the Machine Learning lab) to flag equipments showing signs of imminent failure. Predictive equations may be deployed using:

- PI AF analytics, or
- Using an external script and PI System Access tools such as PI Web API to periodically read data from the PI System, perform the necessary predictive calculations, and finally, use PI Web API to write back the prediction to the relevant AF attribute(s) in the PI System.

In this lab, we show the PI AF Analysis based approach as illustrated by the screen below:



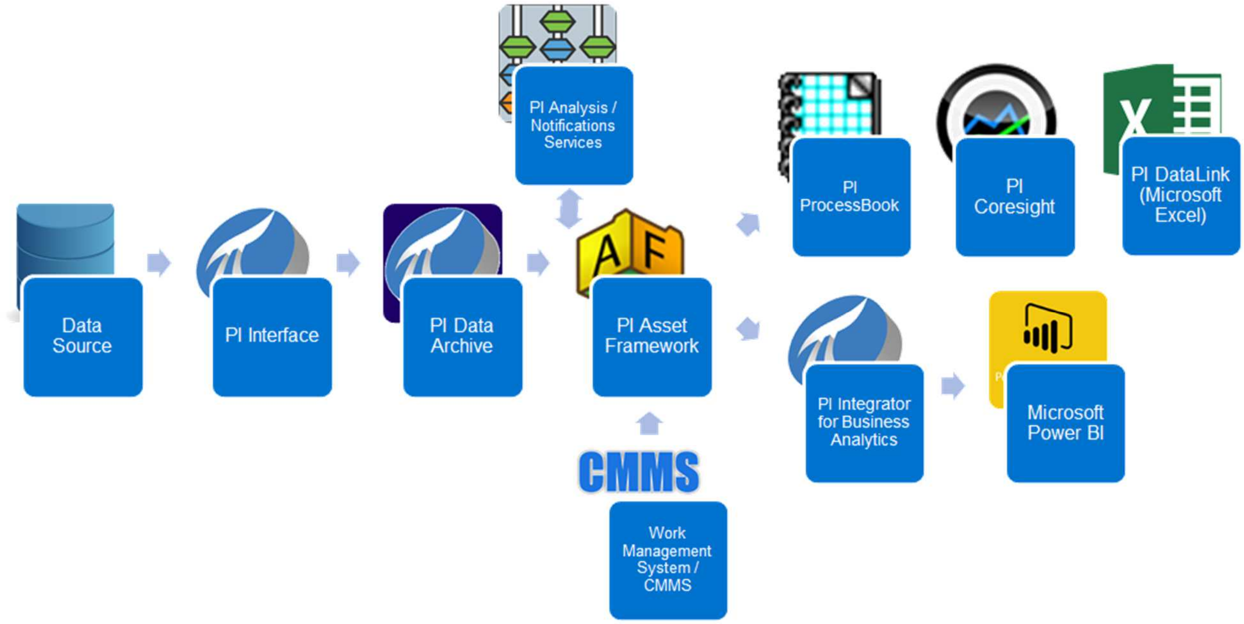
The external script approach is discussed in the Machine Learning Lab notes.

The PI System Architecture for this Lab

In order to complete the following exercises, the PI System will need the following components installed and running:

Software	Recommended Version (used in this lab)
PI Data Archive	2015 R2
PI Asset Framework (PI AF) server	2016
PI Asset Framework (PI AF) client (PI System Explorer)	2016
PI Analysis & PI Notifications Services	2016
PI Coresight	2016
PI DataLink (Microsoft Excel 2013)	2015
PI ProcessBook	2015
PI Integrator for Business Analytics	2016
Microsoft Power BI Desktop	2.31.4280.361

Disclaimer: Several of the PI System Components used in this lab are pre-release versions, and as such some of the user screens may be different from those in the final released software.



1. Time-series sensor data flows into the **PI Data Archive** from a **PI Interface**, itself connected to a **Data Source**.
2. It is then contextualized in **PI Asset Framework** (PI AF). PI AF can also reference external data such as those from **Work Management systems**.
3. The **PI Analysis service** and the **PI Notifications service** perform analyses that use PI AF Attributes as inputs and outputs. The outputs can be stored in PI Tags in the PI Data Archive.
4. **PI ProcessBook, PI Coresight, PI DataLink** are used to visualize PI Server data.
5. **Microsoft Power BI** can be used to visualize data from the PI Server. It can be done in several ways; we recommend using the latest **PI Integrator for Business Analytics** to publish the PI System data in a table format for Business Intelligence (BI) tools.

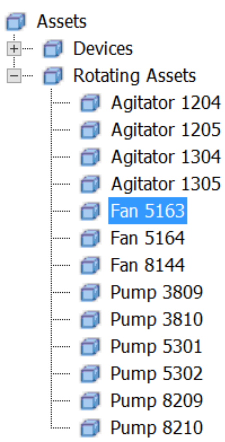
Exercise 1 – Usage-Based Maintenance - Motor Run Hours

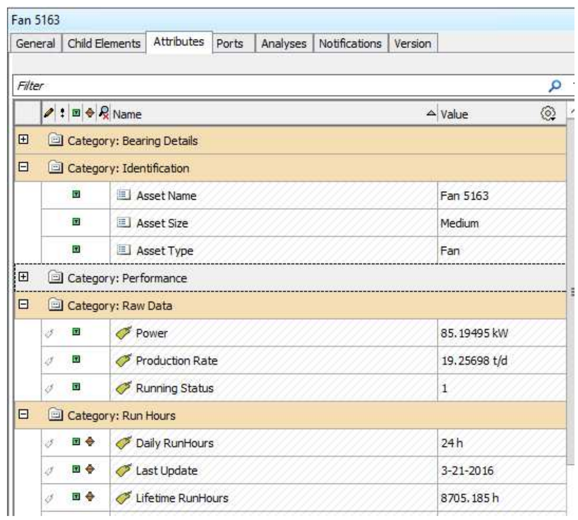
The step-by-step instructions below show how to use the PI Asset Framework (AF) and AF Analysis to calculate motor run hours. This assumes the PI System holds information that allows you to determine when a motor is running, or not running. In this example, the PI Data Archive holds a tag storing the Running Status of the motor (0 = Not running; 1 = Running).

If a status tag is not available, it can be calculated by evaluating the amperage of the motor, or the motor rpm, or another appropriate measurement. For example, IF 'RPM'>=10 THEN "Running" Else "Not Running".

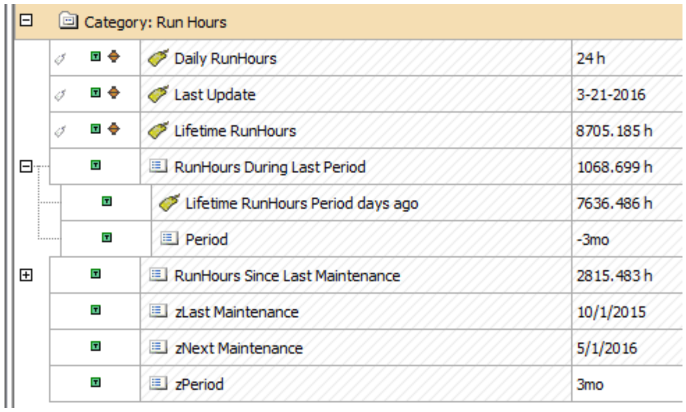
Step 1. Locate the Running Status and RunHours Attributes

1. Open **PI System Explorer**, connecting to the **Energy Management CBM** PI AF database. If the top bar of the window doesn't show \\PISRV01\Energy Management CBM, then click on the top toolbar Database button and select the **Energy Management CBM** database.
2. Navigate to **Fan 5163** asset from the Elements section.

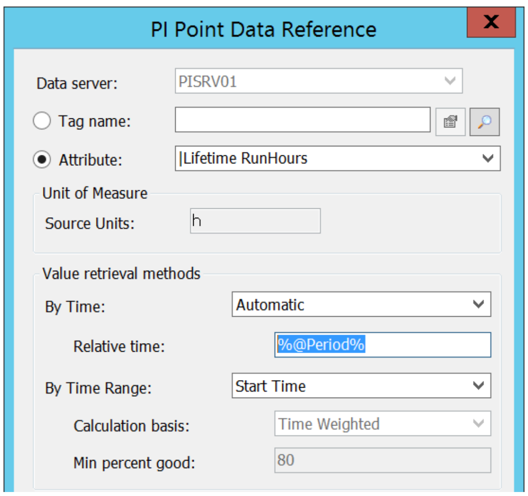




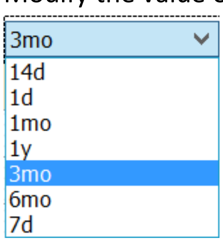
3. Select the Attributes tab, then locate and examine the **Running Status**, **Daily RunHours**, and **Lifetime RunHours** attributes. The Running Status attribute references a raw PI Tag, while the Daily RunHours and Lifetime RunHours are being calculated using a PI AF Analysis (hence the icon shown next to their names).
4. Expand the **RunHours During Last Period** attribute. It is configured as a Formula and subtracts two values of the Lifetime RunHours attribute taken at two different times, defined by the selected **zPeriod** attribute.



Note how the **Lifetime RunHours Period days ago** attribute references the Lifetime RunHours attribute at a point in time, taken from the Period sub-attribute by using a substitution parameter of the form %@Period%:



5. Modify the value of the **zPeriod** attribute by making a selection from the available choices.



6. Refresh the window by using the Refresh top toolbar button, and note the change in the RunHours During Last Period attribute value.
7. Similarly examine how the **RunHours Since Last Maintenance** attribute is being calculated. That should lead the discovery process towards a PI AF Table (accessible from the Library section of PI System Explorer) containing maintenance dates:

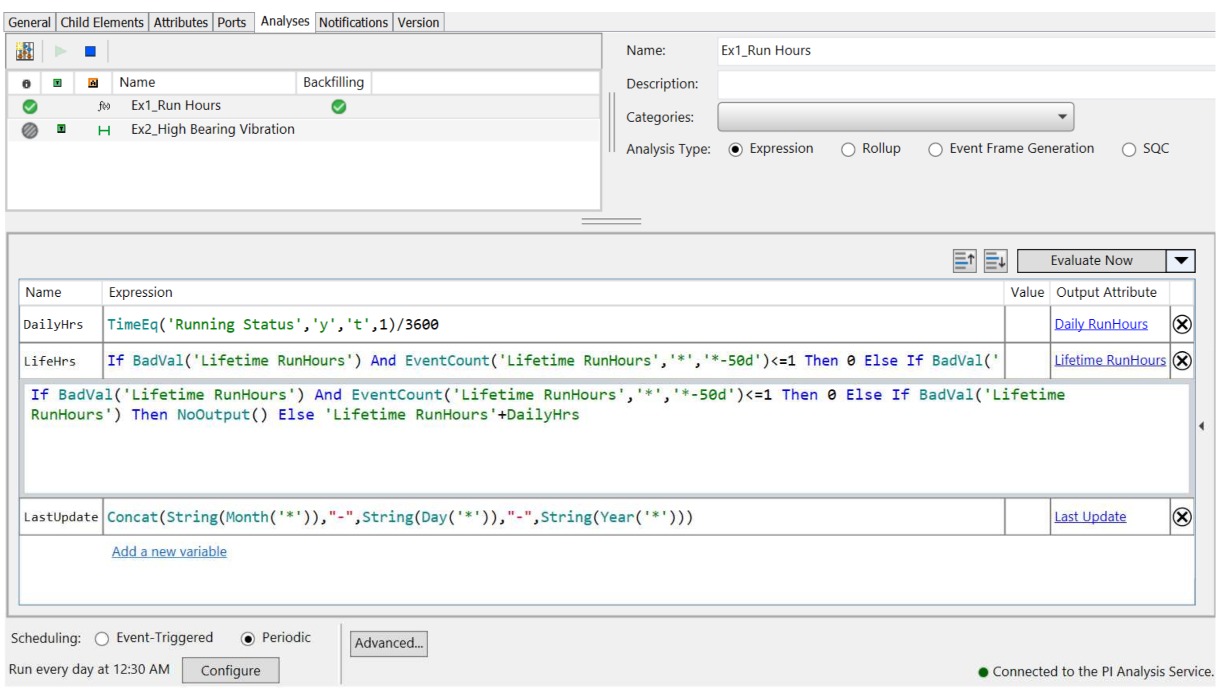
Asset	LastMaintenance	NextMaintenance	AssetType	AssetSize	VibrationLimit
Agitator 1204	1/10/2016	11/10/2016	Agitator	Medium	3
Agitator 1205	2/23/2016	10/3/2016	Agitator	Large	3
Agitator 1304	2/11/2016	12/13/2016	Agitator	Large	3
Agitator 1305	2/11/2016	12/1/2016	Agitator	Medium	3
Fan 5163	10/1/2015	5/1/2016	Fan	Medium	-1
Fan 5164	10/2/2015	5/2/2016	Fan	Medium	-1
Fan 8144	10/5/2015	5/5/2016	Fan	Large	-1
Pump 3809	10/10/2015	5/10/2016	Pump	Small	3
Pump 3810	9/23/2015	7/1/2016	Pump	Small	3
Pump 5301	11/20/2015	8/1/2016	Pump	Medium	3
Pump 5302	11/11/2015	8/23/2016	Pump	Medium	3
Pump 8209	12/2/2015	9/1/2016	Pump	Large	3
Pump 8210	10/2/2015	5/2/2016	Pump	Large	3

These dates are often available in a maintenance/work managment system (such as IBM Maximo, SAP PM, Oracle eAM, Infor etc.), and are referenced in PI AF.

NOTE: The details of integrating maintenance dates with an actual work management systems is outside the scope of this lab. As such, we simply use an AF table containing such dates to illustrate the concept.

Step 2. Examine the Run Hours Analysis

1. Go to the Analyses tab for asset Fan 5163, and select the **Ex1_Run Hours** analysis to examine the variables and functions being used:

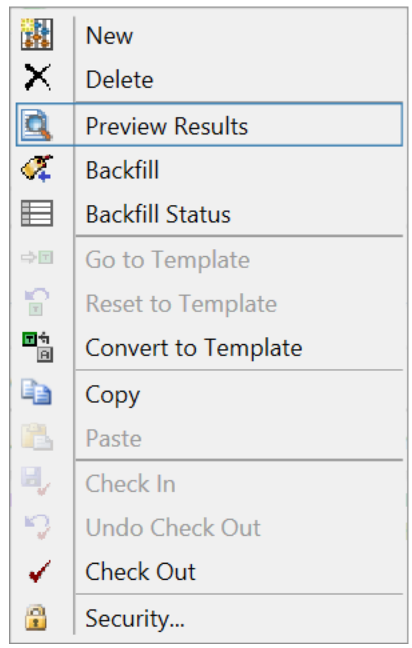


a. The first row calculates the **DailyHrs** variable, storing results to the **Daily RunHours** attribute. It uses the TimeEq() function to calculate the amount of time the Running Status attribute has been equal to 1 (Running) during the past day. Basically, the number of hours the motor has been running in the past day.

b. The second row calculates the **LifeHrs** variable, storing results to the **Lifetime RunHours** attribute. It acts as a totalizer, adding the previously calculated daily hours to the total lifetime run hours. Note the various If-Then-Else statements to ensure the good state of the Lifetime RunHours attribute, before attempting to add the DailyHrs variable to it. The first condition resulting in a 0 value is there to make sure of a successful initialization the first time the expression is executed.

c. Finally, the last row is simply generating a string indicating the last time the expression was executed, and storing it to the **Last Update** attribute. This can be useful information to display on a report.

2. Click the **Evaluate Now** button on the right to see the results that would be stored if the analysis was to be executed right now.
3. Right-click the analysis from the top section and select **Preview Results**. Then enter a time range of interest, for example from **t-10d** to **t**.

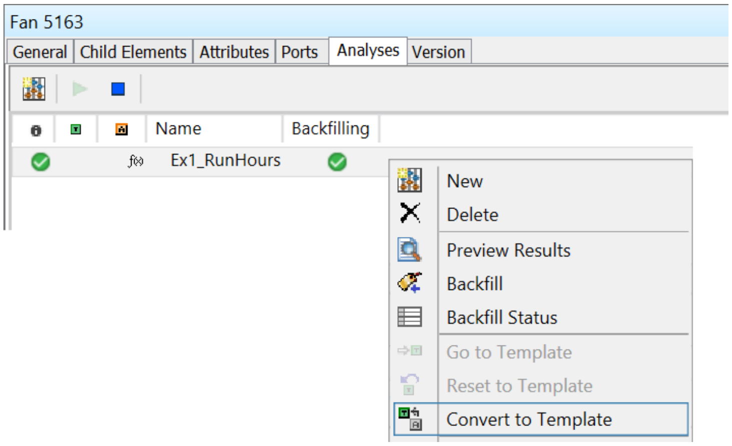


Trigger Time	DailyHrs	LifeHrs	LastUpdate
12/20/2015 12:30:00 AM	22.7	7596.9	12-20-2015
12/21/2015 12:30:00 AM	16.433	7613.4	12-21-2015
12/22/2015 12:30:00 AM	23.133	7636.5	12-22-2015
12/23/2015 12:30:00 AM	24	7660.5	12-23-2015
12/24/2015 12:30:00 AM	19.95	7680.4	12-24-2015
12/25/2015 12:30:00 AM	20.75	7701.2	12-25-2015
12/26/2015 12:30:00 AM	24	7725.2	12-26-2015
12/27/2015 12:30:00 AM	21.183	7746.4	12-27-2015
12/28/2015 12:30:00 AM	23.95	7770.3	12-28-2015
12/29/2015 12:30:00 AM	17.817	7788.1	12-29-2015

4. Note that the analysis has no symbol next to its name, meaning it is not associated with the Element Template yet. This information can be validated by navigating to the Fan 5164 asset and confirming that Ex1_Run Hours analysis does not exist.

Step 3. Deploy the RunHours Analysis to the Whole Fleet

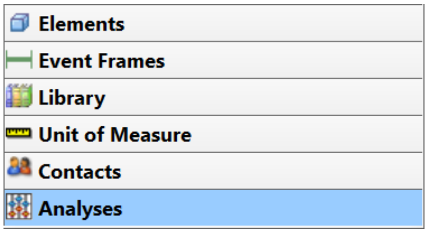
1. Deploy the analysis from Fan 5163 to the whole fleet of motors (fans and pumps) by right-clicking the analysis from the top section and selecting **Convert to Template**.



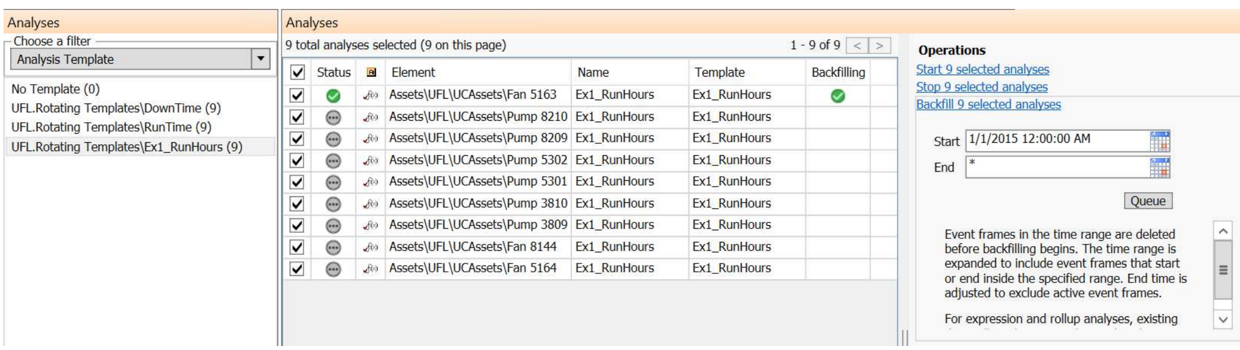
2. Confirm the analysis now exists for other assets, like for **Pump 3809**, and click on the Evaluate button to see the current result.
3. Check in the changes by clicking on the Check In button from the top toolbar.

Step 4. Backfill the Newly Created Analyses

1. Navigate to the **Analyses** main section of PI System Explorer (available from the bottom-left corner of the application).

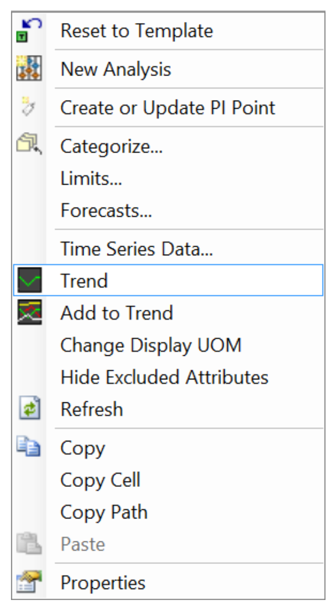


2. Backfill the **Ex1_Run Hours** analyses from this window by following these steps:
    a. Select the Ex1_Run Hours analysis from the left section, by first filtering for Analysis Template
    b. Check all instances of that analysis from the middle section
    c. If not all analyses are showing the ✅ icon, then click the **Start X selected analyses** link from the upper-right section.



d. Finally, launch the backfill process from the top-right section. Backfill the data from January 1st, 2015 as shown below. Click the **Queue** button to launch the backfilling process.

3. Note the progress of the backfill from the bottom-right section.
4. Once completed, confirm the data has been successfully backfilled by selecting the **Fan 5164** from the **Elements** main section of PI System Explorer, right-clicking its **Daily RunHours** attribute and selecting **Trend**.



Step 5. Display the results in PI Coresight

1. A PI ProcessBook display file has been created to display the results. It was published over the Web using the PI Coresight Web application.

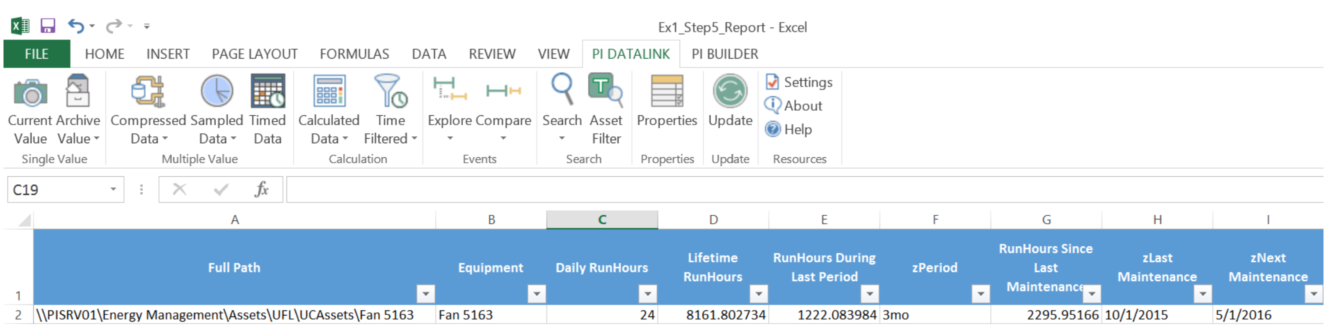
Report - Critical Motors - Run Hours Last Update: 3-12-2016

Equipment	Daily Run Hours	Lifetime Run Hours	During Last Period	Period	Since Last Service	Last Service	Next Service
Agitator 1204	4.51	7,975	0	3mo	387	1/10/2016	11/10/2016
Agitator 1205	23.79	10,119	2,154	3mo	409	2/23/2016	10/3/2016
Agitator 1304	23.49	9,908	2,118	3mo	697	2/11/2016	12/13/2016
Agitator 1305	23.49	9,908	2,118	3mo	697	2/11/2016	12/1/2016
Fan 5163	19.71	8,554	1,174	3mo	2,664	10/1/2015	5/1/2016
Fan 5164	23.97	9,292	2,022	3mo	3,566	10/2/2015	5/2/2016
Fan 8144	14.44	9,839	2,112	3mo	3,635	10/5/2015	5/5/2016
Pump 3809	15.16	8,587	1,949	3mo	3,218	10/10/2015	5/10/2016
Pump 3810	23.97	9,618	2,079	3mo	3,837	9/23/2015	7/1/2016

2. Open **Internet Explorer** from the taskbar, and then use the **PI Coresight** link from the bookmark bar to navigate to the PI Coresight home page.
3. Once on the home page, the **Ex1_Motor Run Hours Report** should be visible. If not, use the left navigation to select ProcessBook Displays >.
4. Modify the time range at the bottom to inspect the values at different times, for instance any day in 2015.

Step 6. Report the Results in Microsoft Excel with PI DataLink (optional)

1. Open the Excel spreadsheet from the machine desktop, named **Ex1_PI DataLink Report.xlsx**.
2. Validate it shows data for all rotating assets.

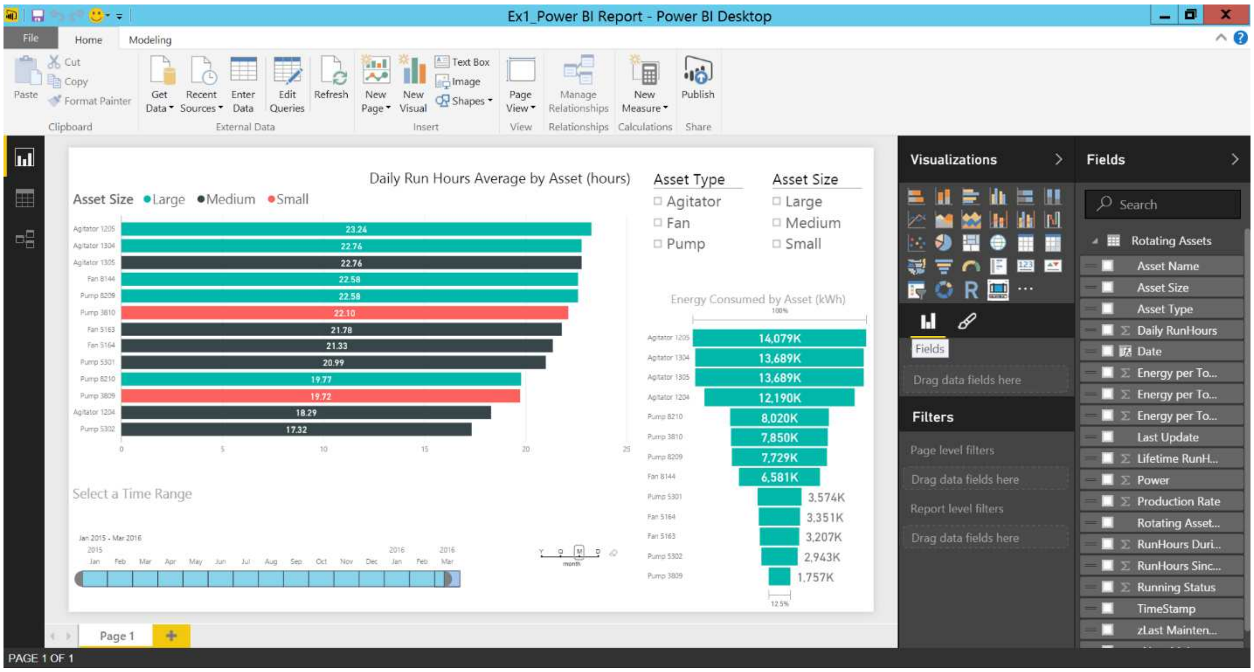


Full Path	Equipment	Daily RunHours	Lifetime RunHours	RunHours During Last Period	zPeriod	RunHours Since Last Maintenance	zLast Maintenance	zNext Maintenance
\\PISRV01\Energy Management\Assets\UFL\UCAssets\Fan 5163	Fan 5163	24	8161.802734	1222.083984	3mo	2295.95166	10/1/2015	5/1/2016

Step 7. Analyze the Results in Microsoft Power BI (optional)

For more details on how to accomplish Power BI reports, please refer to the *Operational Insights using Real-time Dashboards and Self-service Business Intelligence* lab. Here, the PI Integrator for Business Analytics is used to publish the 2015 dataset to be visualized in Power BI Desktop.

1. Open the **Ex1_Power BI Report.pbix** file from the desktop.
2. Use the various slicers/filters to dynamically change the aggregations being performed by this business intelligence tool.



Exercise 2 – Condition-Based Maintenance - Motor Vibration High Condition

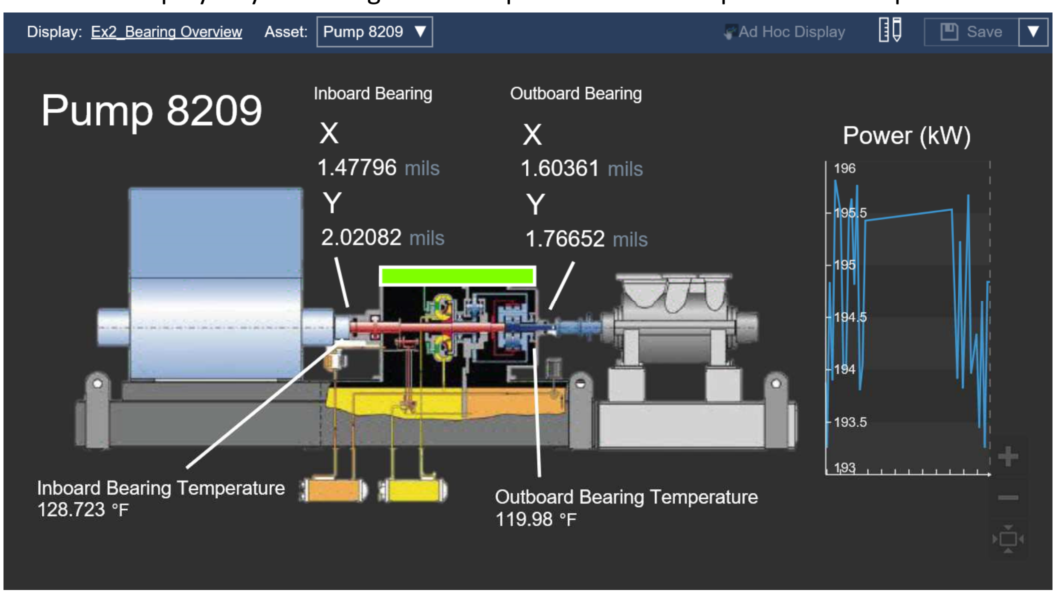
For critical assets, you want to be informed as soon as you as the data you are monitoring indicates a potential malfunction or equipment failure – for example, the two large pumps of this site - **Pump 8209** and **Pump 8210**.

This exercise looks at evaluating a simple condition to determine whether or not a check and/or maintenance should be performed for the asset. Then, it looks at the notification capability of the PI System, more specifically at sending an email notification if an action needs to be taken for the pumps. The bearings are the pieces that will be evaluated in the PI Server for those pumps, and specifically vibration and temperature measurements will be used to assess the equipment's condition.

More complex logic/rule to determine the asset's condition could be performed in the PI Server, or even externally. Exercise 3 discusses how you can use external tools with PI System data to perform advanced analytics/machine learning and bring back the results to the PI System for predictive maintenance.

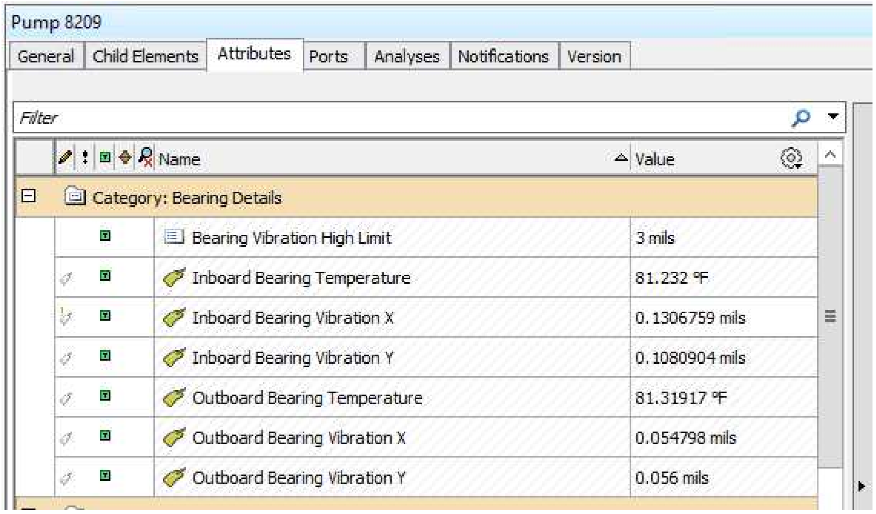
Step 1. Examine the Available Information in PI AF

1. Open **Internet Explorer**, navigate to the PI Coresight home page and open the **Ex2_Bearing Overview** display. Try switching the Asset pick list from Pump 8209 to Pump 8210.



2. Navigate to element **Pump 8209** in PI System Explorer, and look at its Attributes.

3. Expand the **Bearing Details** attribute category. Available are information on the following:



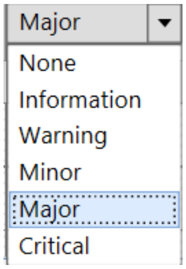
There are two bearings for each pump, one inboard and one outboard, with each having two sensors for capturing movements on the X and Y axis. Bearing temperature in degrees Fahrenheit is also available.

Step 2. Confirm the Event Frame Generation Analysis is Working

1. Each one of the bearing vibration measurement is being evaluated against the Bearing Vibration High Limit. Event frames are being generated when values go above that limit. From the Analyses tab, select the **Ex2_High Bearing Vibration** analysis and take a look at the **Start trigger**, as well as the **Variables** definition used to simplify the syntax.

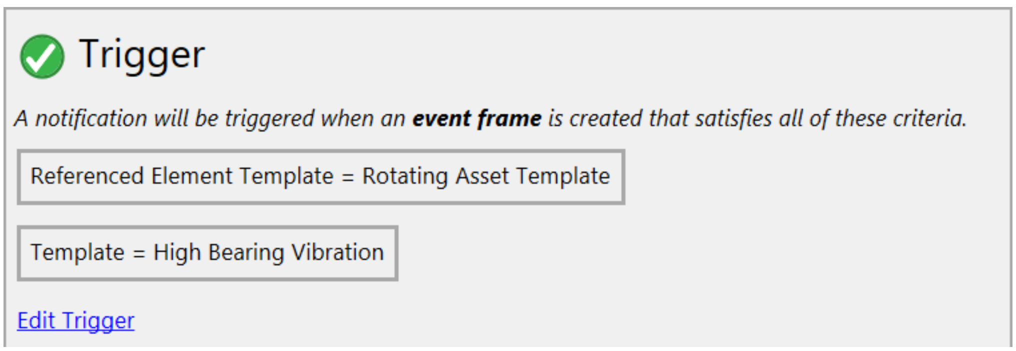
Name	Expression
⊟ Variables	
Limit	'Bearing Vibration High Limit'
InXFault	'Inboard Bearing Vibration X'>Limit
InYFault	'Inboard Bearing Vibration Y'>Limit
OutXFault	'Outboard Bearing Vibration X'>Limit
OutYFault	'Outboard Bearing Vibration Y'>Limit
⊟ Start triggers	
StartTrigger1	If InXFault Or InYFault Or OutXFault Or OutYFault Then True Else False
⊟ End trigger	
EndTrigger	*Type an expression (optional)*

2. Click the **Evaluate Now** button to verify if the trigger is currently active or not.
3. Right-click the analysis name from the top section and select Go to Template . In the analysis template, the **Severity** drop-down can be used to categorize different events based on their importance.

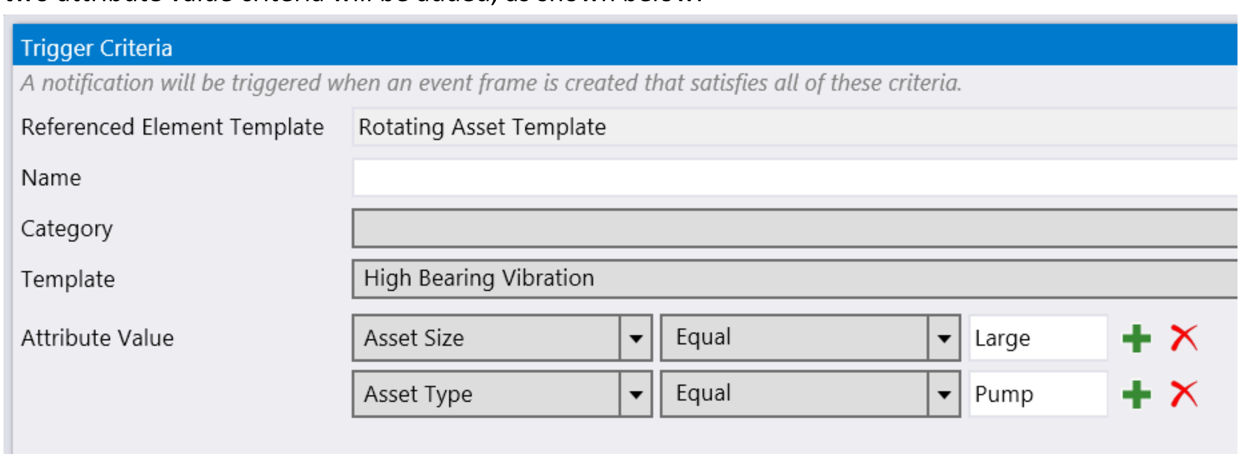


Step 3. Finalize the Configuration of the Notification Trigger Rule

1. Optionally, it is now possible with PI AF 2016 to enable the notification feature on top of an Event Frame. To access this option, select the **Notification Templates** tab for the Rotating Asset Template.
2. For the High Bearing Vibration Notification Template, click on the **Edit Trigger** link from the Trigger section.



3. Since notifications are only desired for the two large pump assets, Pump 8209 and Pump 8210, two attribute value criteria will be added, as shown below:

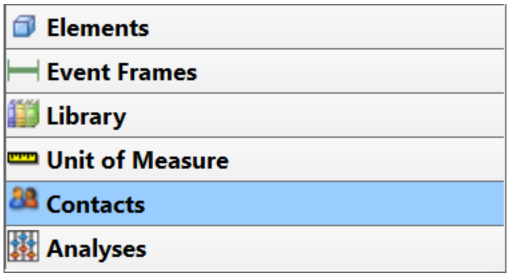


Hence only when event frames will be generated for pumps 8209 or 8210 will it also trigger a notification.

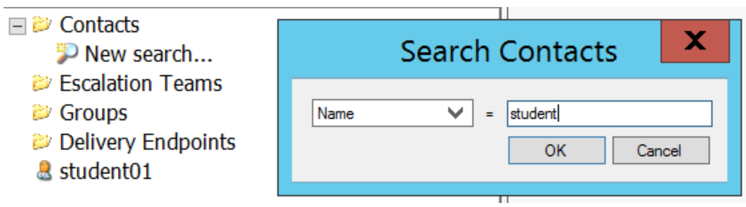
4. Save the changes by clicking on the Check In button from the top toolbar.

Step 4. Customize the Student Contact Email Address

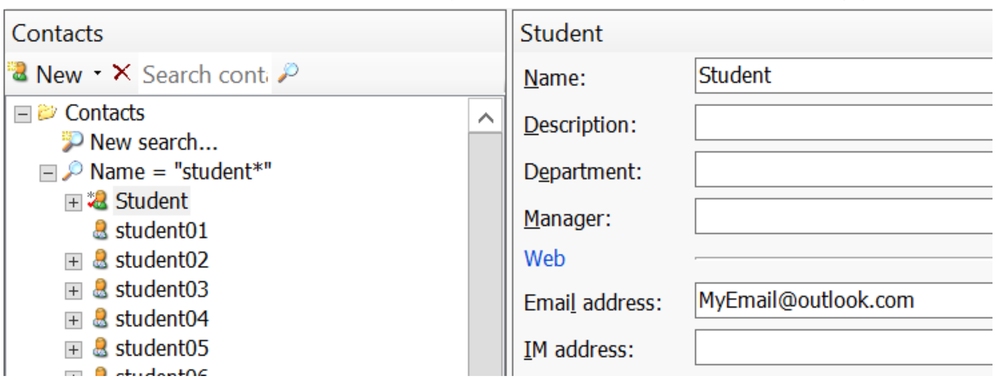
1. Since the goal is to receive an email notification when the vibration is too high, a preferred email address should now be chosen to test the email delivery. Navigate to the **Contacts** section of PI System Explorer.



2. Expand the **Contacts** folder, click on **New search…**, and search for the contact named **Student**.



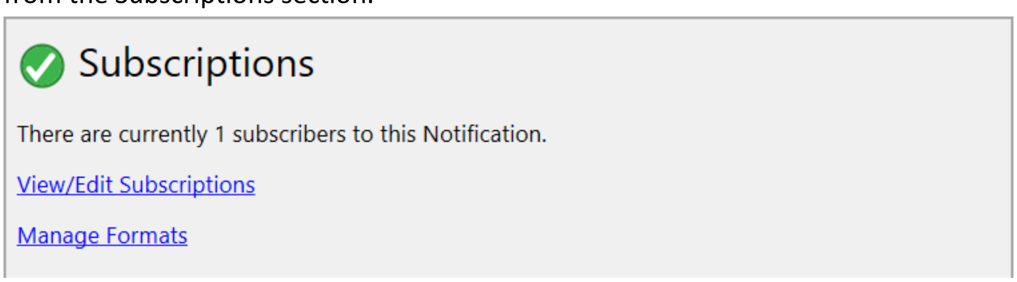
3. Select the Student contact and modify its **Email address** field accordingly.



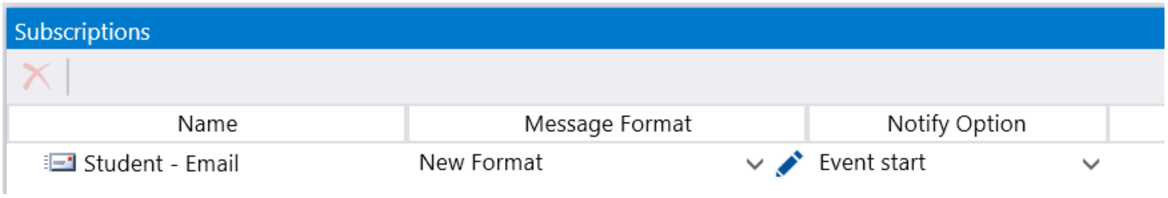
4. Check In the changes.

Step 5. Confirm your Subscription to the Email Notification

1. Navigate back to the **Library** section of PI System Explorer.
2. For the High Bearing Vibration notification template, click on the View/Edit Subscriptions link from the Subscriptions section.



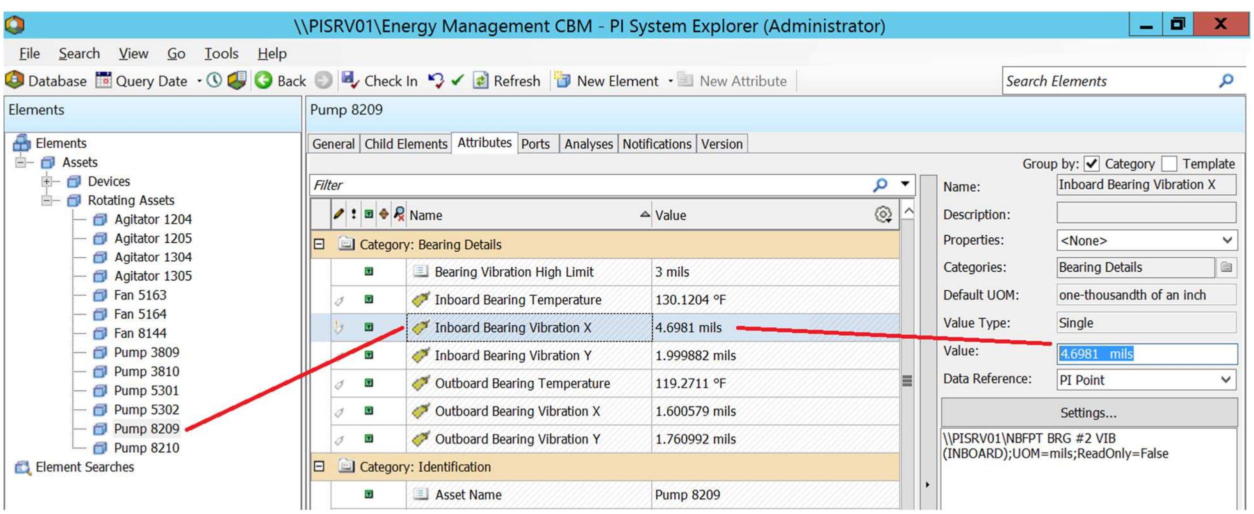
3. Confirm the **Student - Email** entry is present, meaning a triggered notification would send an email to the configured email address for that contact. Note the Notify Option can be set to send an email both at the beginning and the end of an alert.



4. The pencil icon ( ) next to the Message Format column can be used to modify the message formatting to be sent via email. This is useful if the same alert needs to be sent to two different contacts in the enterprise with different roles, hence not necessarily requiring the same amount of details on a particular event. For the purpose of this lab, the message format already selected will be used.

Step 6. Force the Triggering of the Notification Rule

1. The trigger rule in this case will generate an event frame whenever one of the vibration reading goes above the limit of 3 mils. In order to test the email delivery, the Inboard Bearing Vibration X attribute will be manually sent a value above that limit. First navigate to the **Elements** section of PI System Explorer.
2. Select the **Pump 8209** asset and its **Inboard Bearing Vibration X** attribute.
3. From the configuration pane on the right, highlight the **Value:** field. Entering a value in that field and hitting the **Enter** key on the keyboard will cause that entered value to be sent to the PI Tag, and will trigger a notification if that value is above 3 mils. Enter a value greater than 3 in that field and hit Enter.

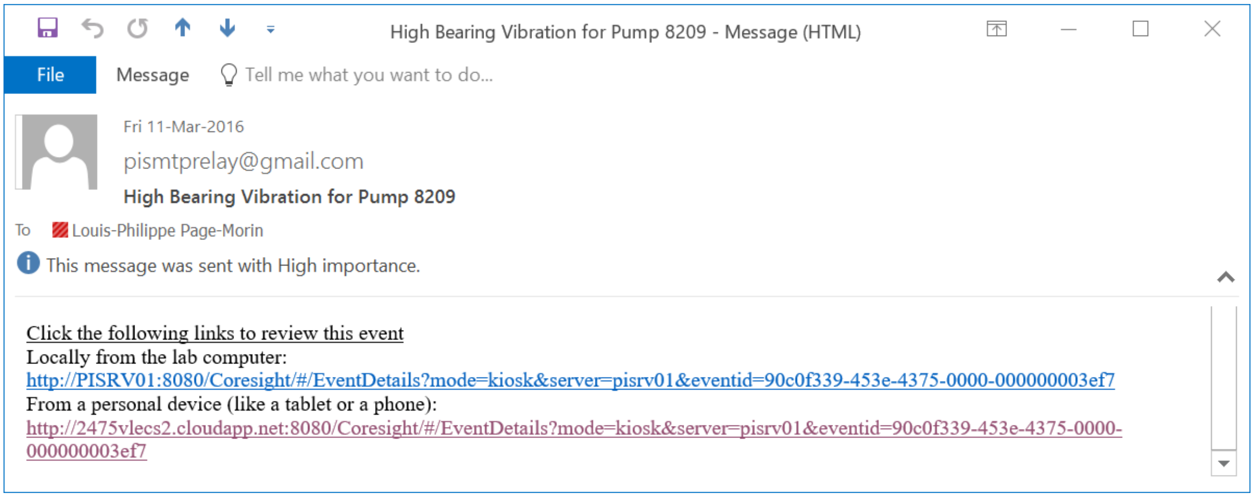


4. Wait a few seconds and an email should be delivered to the previously configured email address.

Step 7. Check the Notification in PI Coresight

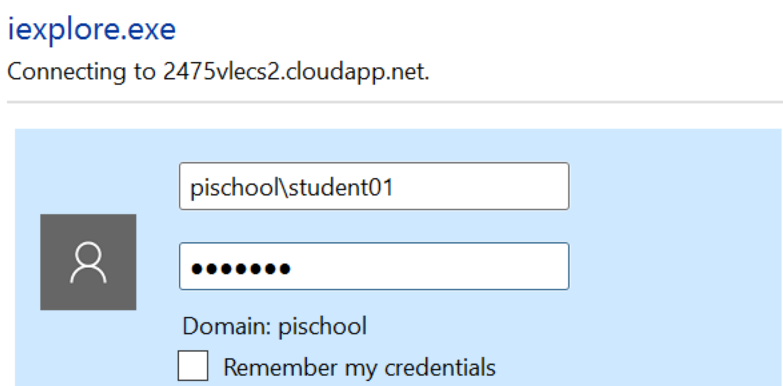
1. Once the email is received, notice the information displayed, and specifically check for the two links at the bottom. Both server the same purpose, that is to allow an end user to review the alert in the PI Coresight Web application. One allows to do this within the lab VM environment, the other one is to access PI Coresight from the outside world, say on a tablet, phone, or any

personal device connected to the internet. Click the appropriate link to access the PI Coresight page.

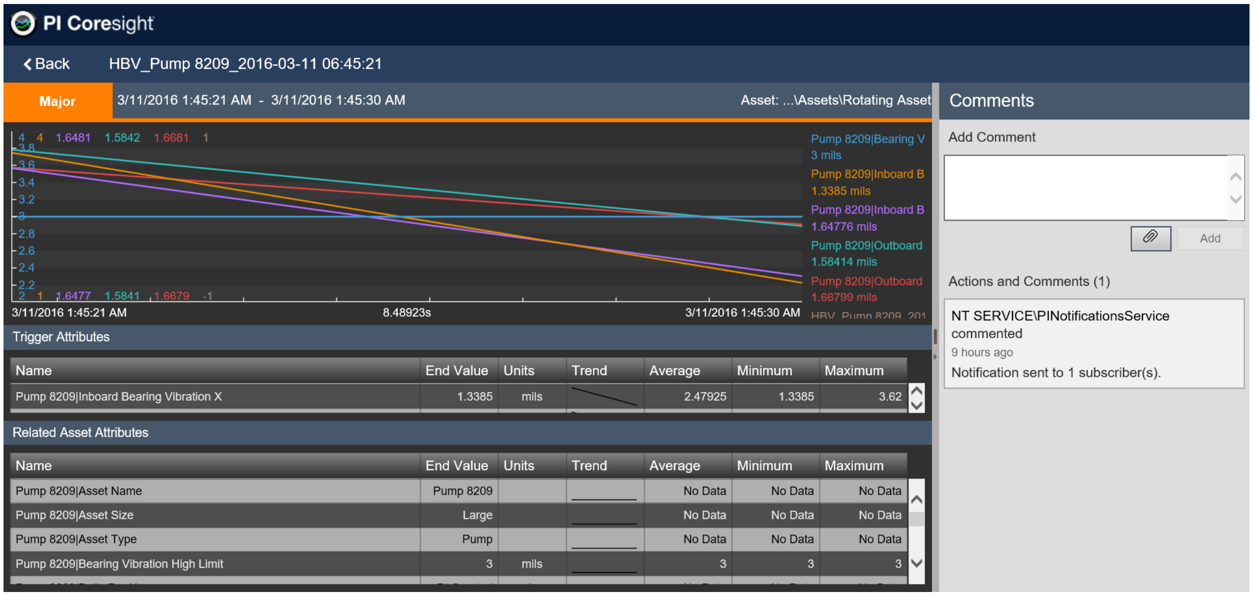


IMPORTANT: The lab VM machine name will be different for every student, please make sure to edit the machine name, here shown as **2475**vlecs2.cloudapp.net, with the machine name that was assigned to you.

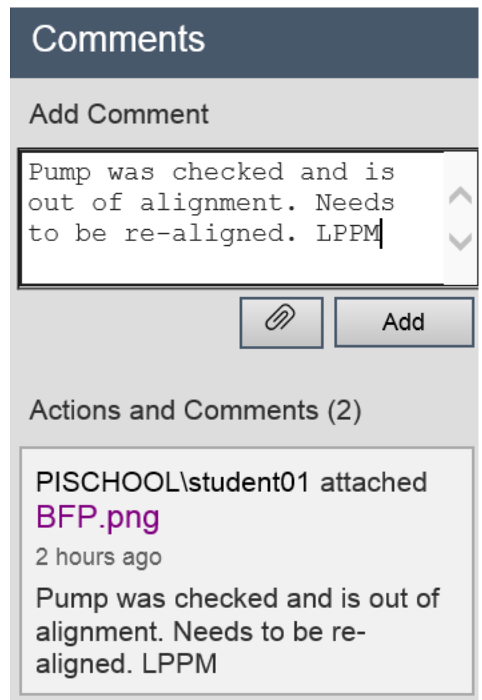
2. If asked for credentials, use the **pischool\student1** user with the **student** password to login.



3. From the PI Coresight page that opens up, examine the top trend and the bottom attributes table giving the measurement details within the time frame of the notification.



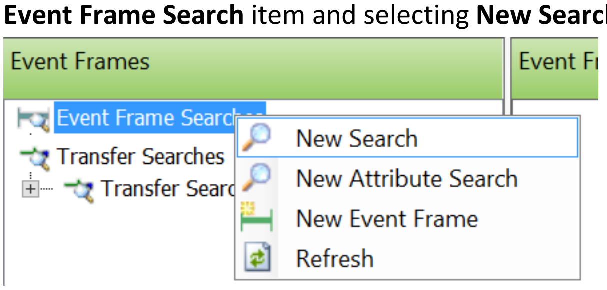
Step 8. Comment/Annotate the Notification in PI Coresight



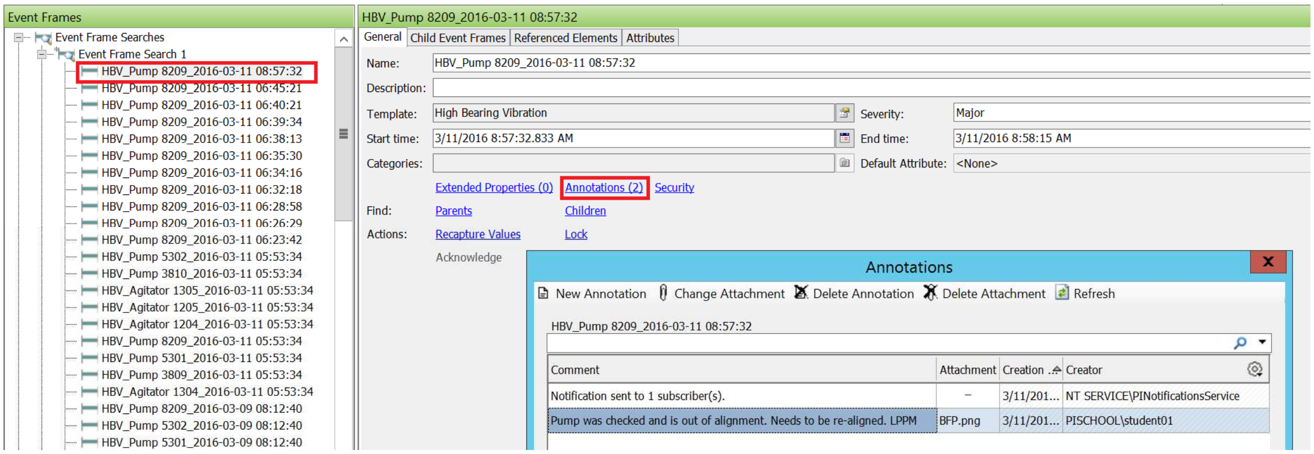
1. From the same PI Coresight screen, add a comment for the Notification, say for example:
   "Pump was checked and is out of alignment. Needs to be re-aligned. LPPM"
2. Try to add a file to the event by clicking the 📎 button. Depending on the device being used to access the PI Coresight page, it will be possible to browse the device for files or pictures, or even to open the camera of the mobile device automatically to take a picture or a video/audio file, and save it to the notification event.
3. Save the comment and/or attachment by clicking the **Add** button.

Step 9. Confirm the Addition of the Comment to the Event Frame

1. Navigate to the **Event Frames** section of PI System Explorer.
2. Search for the latest event frames by first right-clicking the **Event Frame Search** item and selecting **New Search**.



3. From the window that opens up, leave the default settings, press the Search button in the top-right section, and then the OK button in the bottom-right section.
4. From the results that are displayed, find the most recent High Bearing Vibration event for Pump 8209, select it from the left tree view, and navigate to its **General** tab.



5. Click the **Annotations** link and confirm the entries that were just added are showing up as expected.

Exercise 3 – Predictive Maintenance – Engine Failure

In this Exercise, we review the predictive model created in *Use Data Science for Machine Learning and Predictions based on PI System data* lab (hereafter referred as the Machine Learning lab). The model is used to flag equipments showing signs of imminent failure. Predictive equations may be deployed in the PI System using:

- PI AF analytics, or
- With an external script and PI System Access tools such as PI WebAPI to periodically read data from the PI System, perform the necessary predictive calculations, and finally, use PI WebAPI to write back the prediction to the relevant AF attribute(s).

The PI AF analytics screens are shown below. The external script approach is discussed in the Machine Learning lab notes. The Machine Learning lab summary is as below:

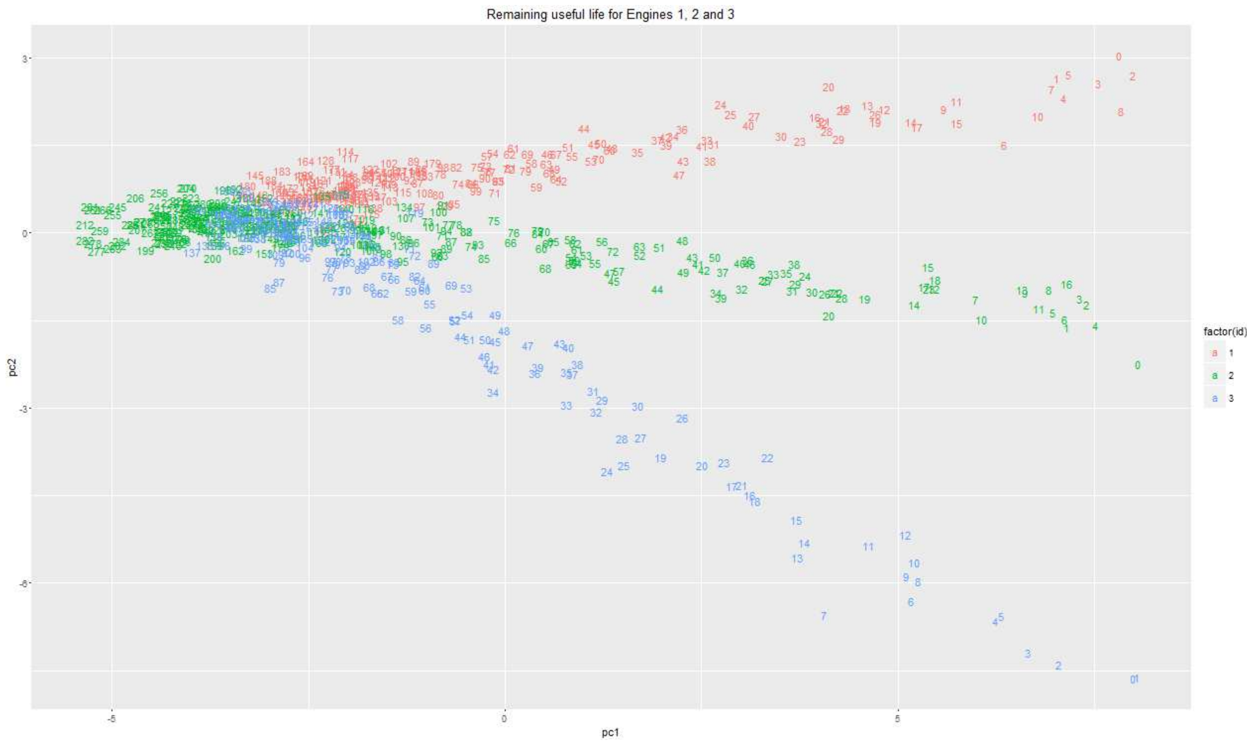
Machine Learning Lab Exercise: Predict imminent equipment failure using sensor data and failure history

In a deployment with 100 similar engines, sensor data such as rpm, burner fuel/air ratio, pressure at fan inlet, and twenty other measurements plus settings for each engine – for a total of about 2000 tags – are available. On average, an engine fails after 206 cycles, but it varies widely - from about 130 to 360 cycles.

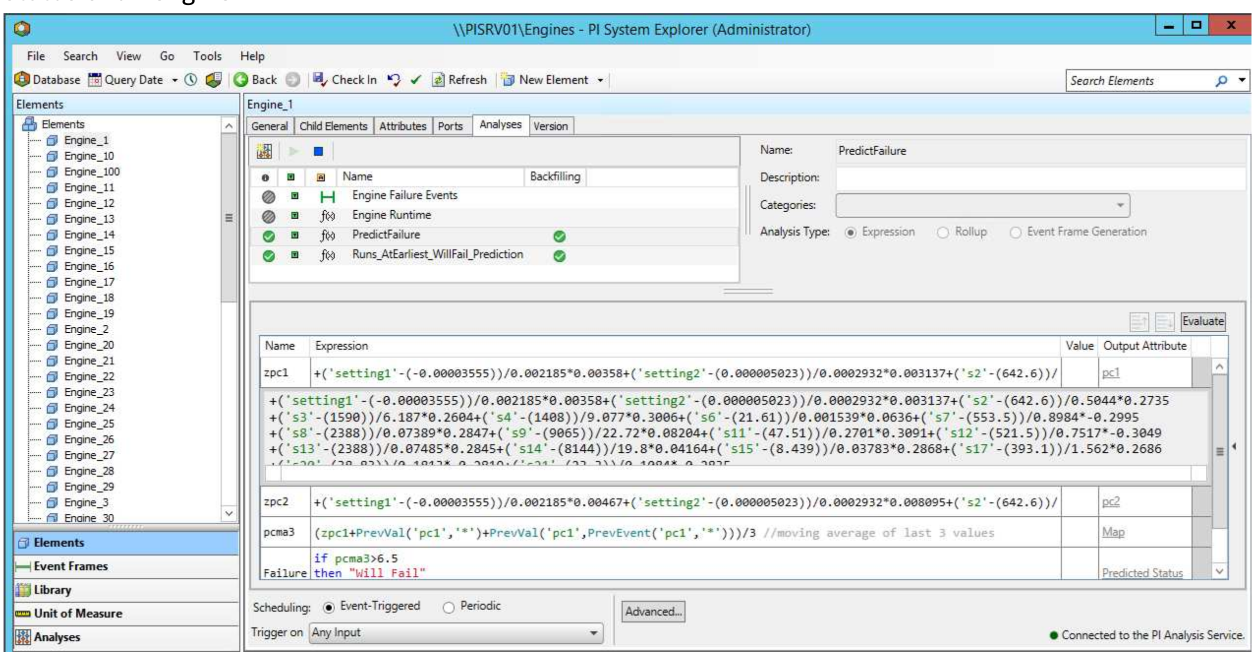
Using an open source machine learning tool such as R, you will create a multivariate model to predict engine failures within approximately a 15 cycle window ***before they fail***. The lab will walk through the end-to-end data science process – preparing the dataset, visually exploring it, partitioning the data for training and testing, validating the models using previously unseen data, and finally deploying the model using the PI System for predictive maintenance.

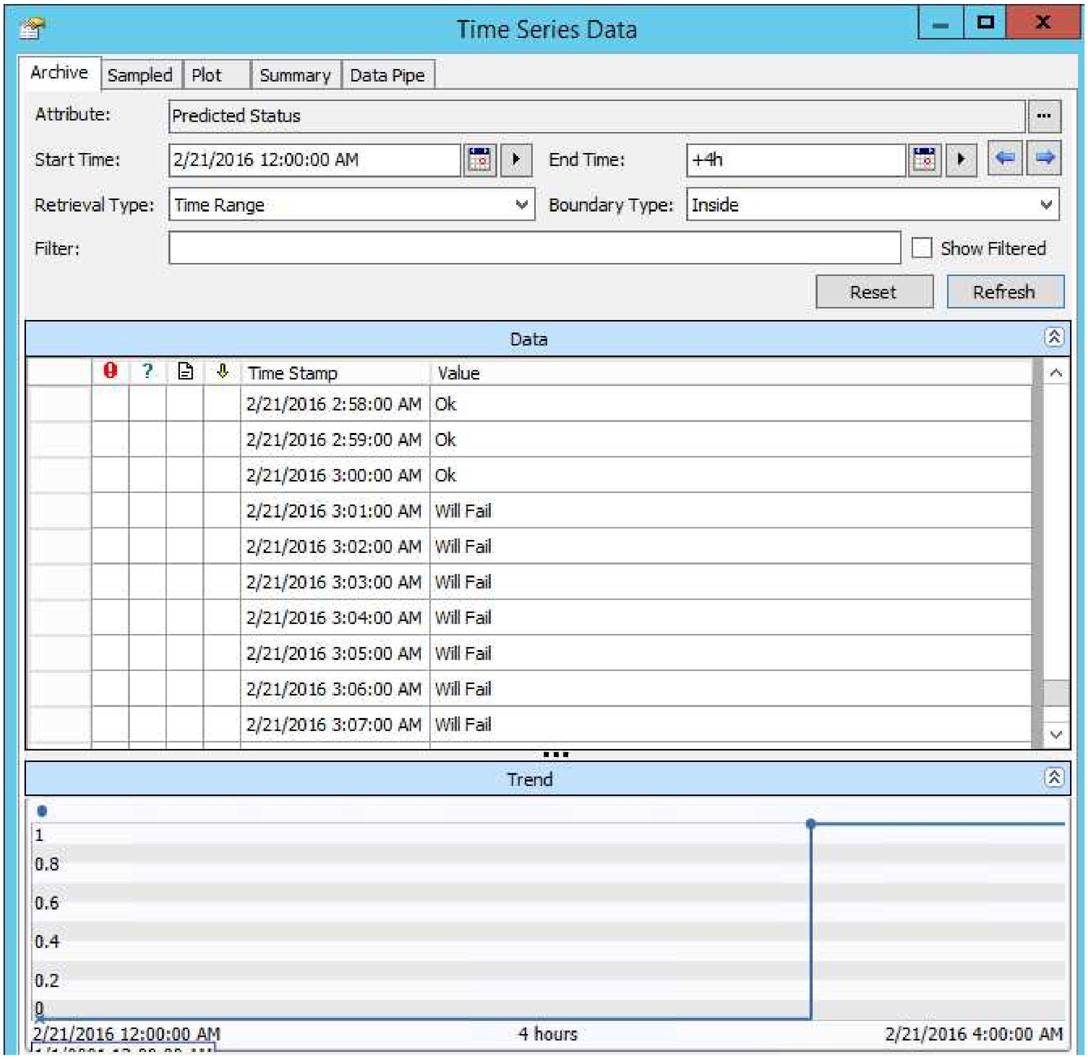
Level: 300 (familiarity with R will be useful but is not a requirement)

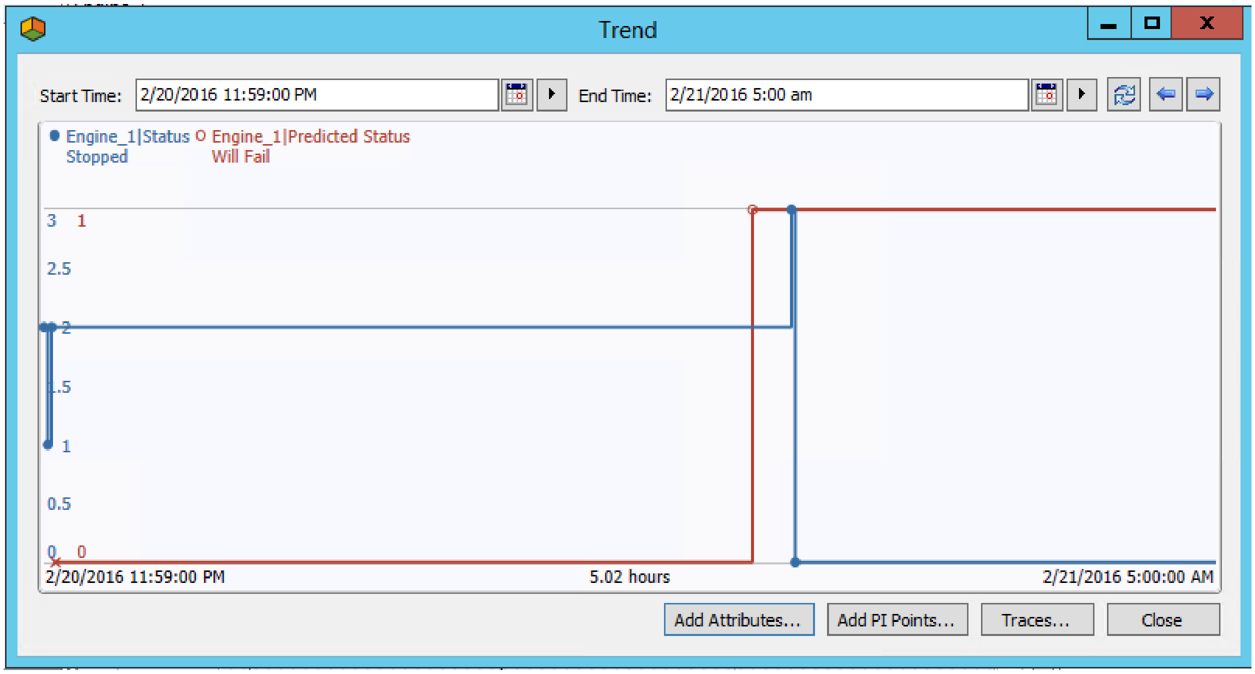
	A	B	C	D	E	F	G	H	I	J	K	L	M
2	id	cycle	setting1	setting2	setting	s1_T2_FanIn	s2_T24_LPCOut	s3_T30_HPCO	s4_T50_LPTC	s5_P2_FanIn	s6_P15_Bypa	s7_P30_HPCOu	s8_Nf_FanRPM
3	1	1	-0.0007	-0.0004	100	518.67	641.82	1589.7	1400.6	14.62	21.61	554.36	2388.06
4	1	2	0.0019	-0.0003	100	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04
191	1	189	-0.0006	0.0002	100	518.67	644.18	1596.17	1428.01	14.62	21.61	550.7	2388.27
192	1	190	-0.0027	0.0001	100	518.67	643.64	1599.22	1425.95	14.62	21.61	551.29	2388.29
193	1	191	0	-0.0004	100	518.67	643.34	1602.36	1425.77	14.62	21.61	550.92	2388.28
194	1	192	0.0009	0	100	518.67	643.54	1601.41	1427.2	14.62	21.61	551.25	2388.32
195	2	1	-0.0018	0.0006	100	518.67	641.89	1583.84	1391.28	14.62	21.6	554.53	2388.01
196	2	2	0.0043	-0.0003	100	518.67	641.82	1587.05	1393.13	14.62	21.61	554.77	2387.98
197	2	3	0.0018	0.0003	100	518.67	641.55	1588.32	1398.96	14.62	21.6	555.14	2388.04
198	2	4	0.0035	-0.0004	100	518.67	641.68	1584.15	1396.08	14.62	21.61	554.25	2387.98
199	2	5	0.0005	0.0004	100	518.67	641.73	1579.03	1402.52	14.62	21.6	555.12	2388.03
200	2	6	-0.001	0.0004	100	518.67	641.3	1577.5	1396.76	14.62	21.61	554.98	2388.04



The final step in the Machine Learning lab uses PI AF to deploy the predictive equation via AF analysis to predict "OK" or "Will Fail" status. The screens below show the AF analysis and the predicted and actual Status of an engine.




Time Series Data
Archive
Sampled
Plot
Summary
Data Pipe
Attribute:
Predicted Status
Start Time:
2/21/2016 12:00:00 AM
End Time:
+4h
Retrieval Type:
Time Range
Boundary Type:
Inside
Filter:
Show Filtered
Reset
Refresh
Data
Time Stamp
Value
2/21/2016 2:58:00 AM
Ok
2/21/2016 2:59:00 AM
Ok
2/21/2016 3:00:00 AM
Ok
2/21/2016 3:01:00 AM
Will Fail
2/21/2016 3:02:00 AM
Will Fail
2/21/2016 3:03:00 AM
Will Fail
2/21/2016 3:04:00 AM
Will Fail
2/21/2016 3:05:00 AM
Will Fail
2/21/2016 3:06:00 AM
Will Fail
2/21/2016 3:07:00 AM
Will Fail
Trend
1
0.8
0.6
0.4
0.2
0
2/21/2016 12:00:00 AM
4 hours
2/21/2016 4:00:00 AM



Note that AF predicts that the engine "Will Fail" (red trace) before actual failure (blue trace).

Please consult the "Use Data Science for Machine Learning and Predictions based on PI System data" lab notes for additional information.