PI World 2019 Lab

Aligning asset maintenance with operations – failure modes, usage-based, condition-based and predictive (pattern recognition) maintenance

OSIsoft, LLC
1600 Alvarado Street
San Leandro, CA 94577 USA
Tel: (01) 510-297-5800
Web: http://www.osisoft.com

Published: March 21, 2019

Maintenance and Reliability – CBM/Condition Monitoring – Hands-on Lab – OSIsoft PI World 2019

Lead: Gopal GopalKrishnan, P.E., Solution Architect, Partners & Strategic Alliances

Lead: Keith Pierce, Technical Advisor, Delivery Services

Instructor: Bryan Pope, Systems Engineer

Instructor: Mike Santucci, ECG Inc. (OSIsoft Partner)

Instructor: Preston Johnson, Allied Reliability (OSIsoft Partner)

Table of Contents

Aligning asset maintenance with operations – failure modes, usage-based, condition-based and predictive (pattern recognition) maintenance

Lab Description

Increasing equipment uptime means preventing failures before they happen; and in turn, this requires you to have a list of likely failures and the appropriate condition monitoring for the process or equipment/component. Attend this lab to learn about failure modes, and the corresponding monitoring techniques to prevent failures. The lab will also cover the use of operations data for a layered approach to uptime and reliability via usage based, condition-based and predictive (pattern recognition based) maintenance.

Usage-based maintenance includes using operational metrics such as motor run-hours, compressor starts/stops, grinder tonnage etc. And, condition-based maintenance utilizes measurements such as filter deltaP, bearing temperature, valve stroke travel, and others. Predictive maintenance can be simple predictive such as monitoring vibration (rms, peak etc.) to predict RUL (remaining useful life) or heat-exchanger fouling to schedule cleaning. Advanced predictive maintenance use cases include pattern recognition or other machine learning techniques for detecting anomalies/predicting failures.

Who should attend? Power User and Intermediate

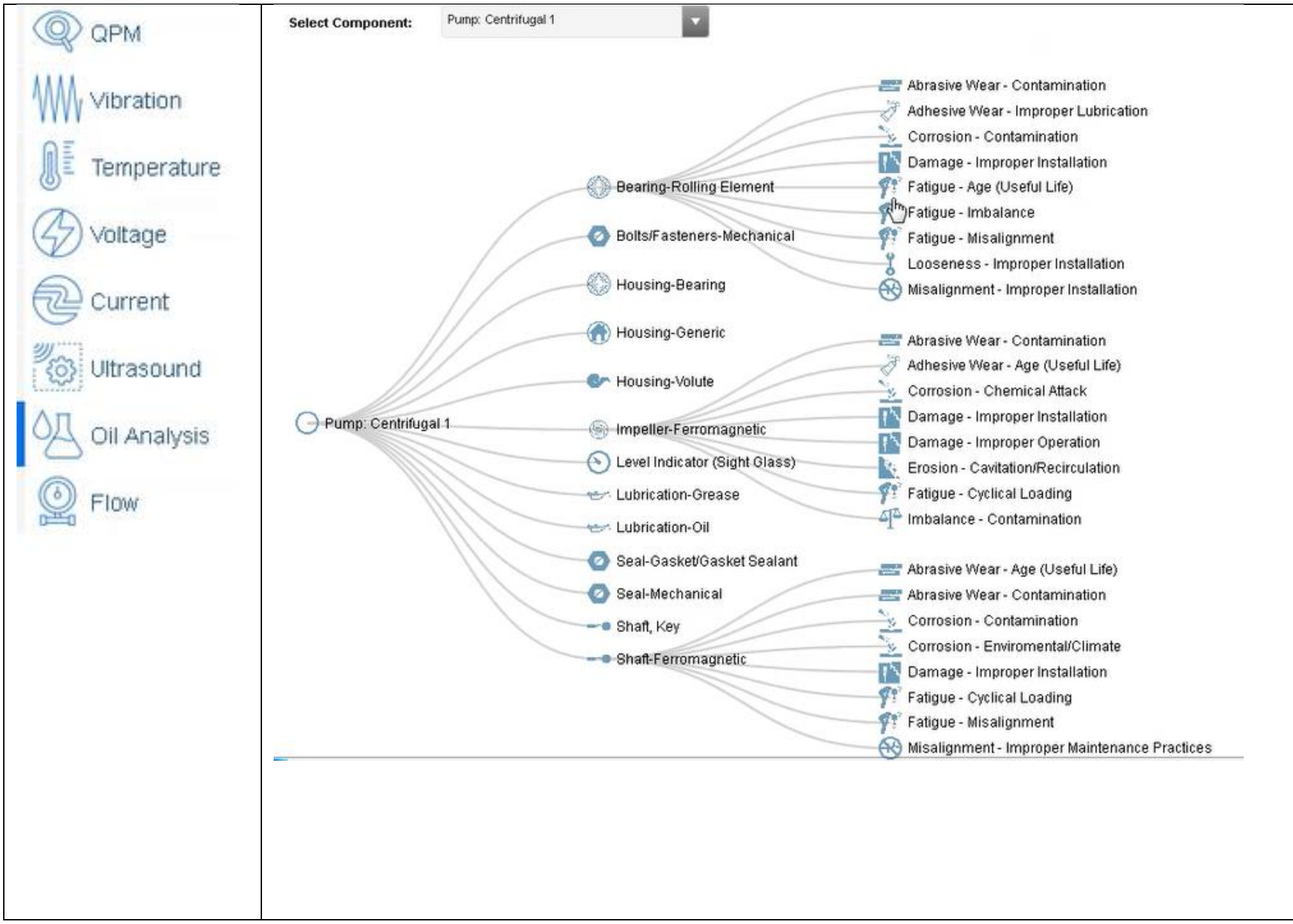
Duration: 3 hours

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 broader definition. This is in contrast to a break-fix strategy (reactive maintenance), and calendar scheduled maintenance (clean and lube every 3 months, laser align every 6 months etc.) regardless of the condition of the asset and whether it was used or not.

Increasing equipment uptime means preventing failures before they happen; and in turn, this requires you to have a list of likely failures and the appropriate condition monitoring for the process or equipment/component.

As such, we begin with a review of failure modes (see figure below) for a commonly used equipment - pump/motor, and the existing sensor measurements you may already have for this pump/motor and any additional sensor coverage and condition monitoring that will be required to prevent certain failure modes. We will discuss the use of both process data and machine condition data for CBM and failure prevention.



And, then we will cover the use of equipment and process data for a ***layered approach*** to uptime and reliability via usage based, condition-based and predictive (pattern recognition based) maintenance.

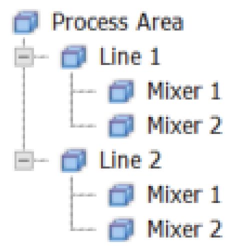
- Exercise 1: Usage base maintenance – motor run-hours and valve actuation counts
- Exercise 2: Condition-based maintenance – bearing temperature high alert
- Exercise 3a: Predictive maintenance (simple) – univariate (single variable) – decreasing compressor efficiency trend extrapolated to predict time to maintenance
- Exercise 3b: Predictive maintenance (advanced) – multivariate use case – pulverizer – early fault detection for the bearing based on APR (advanced pattern recognition) analytics
- Exercise 4: Asset health score – you utilize multiple condition assessment rules with appropriate weighting factors to process/equipment measurements to calculate an overall asset health score

You may also find it useful to review related content from CBM Hands-on Labs in 2016 and 2017 .

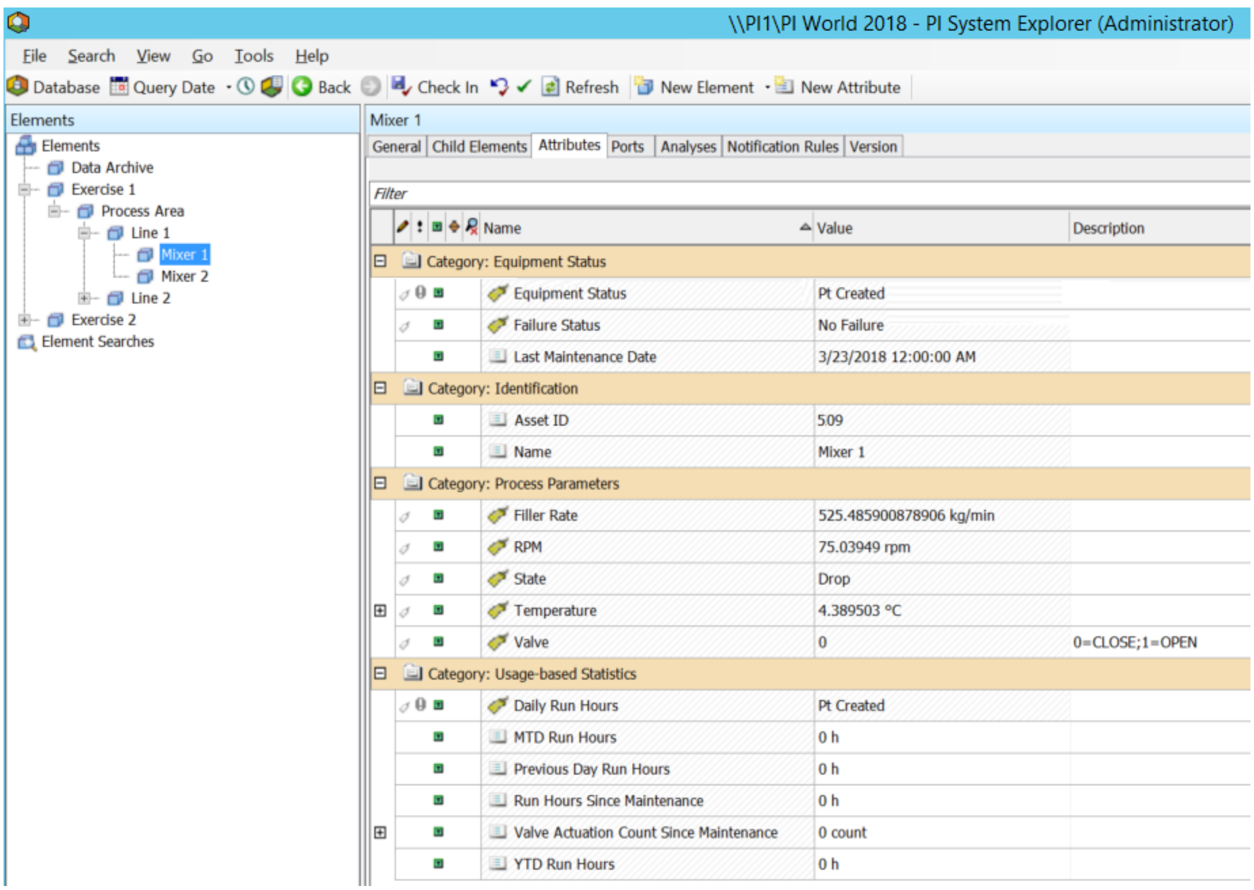
Exercise 1: Usage-based Maintenance (UbM)

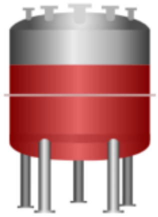
In this exercise, motor run-hours and valve actuation counts are calculated to serve as a basis for usage-based maintenance.

We use an ice-cream factory running two process lines – Line 1 and Line 2, with two mixers on each line.



The hands-on portion includes building the run-hours calculations in AF, and the relevant PI Vision displays as shown below.

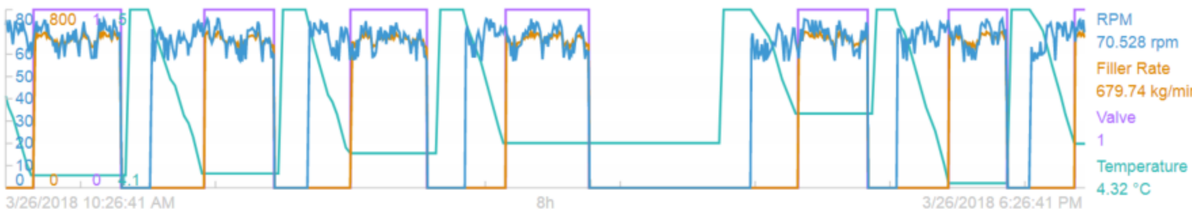


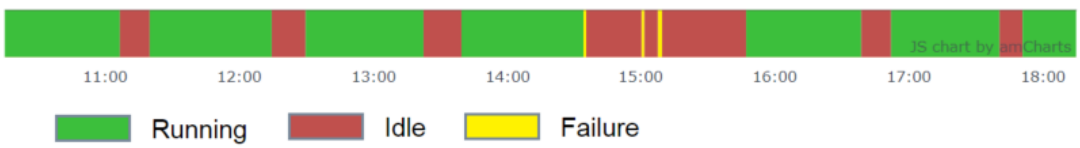


Mixer 1

509

Name	Value
Mixer 1|Last Maintenance Date	3/26/2018 12:00:00 AM
Mixer 1|MTD Run Hours	55.643
Mixer 1|Previous Day Run Hours	16.244
Mixer 1|Run Hours Since Maintenance	14.827
Mixer 1|Valve Actuation Count Since Maintenance	16
Mixer 1|YTD Run Hours	55.643





Name	Value	Units
Mixer 1|Last Maintenance Date	3/26/2018 12:00:00 AM	
Mixer 1|MTD Run Hours	55.143	h
Mixer 1|Previous Day Run Hours	16.244	h
Mixer 1|Run Hours Since Maintenanc	14.327	h
Mixer 1|Valve Actuation Count Since	16	count

Name	Value	Units
Mixer 1|Last Maintenance Date	3/23/2018 12:00:00 AM	
Mixer 1|MTD Run Hours	7.5494	h
Mixer 1|Previous Day Run Hours	0	h
Mixer 1|Run Hours Since Maintenanc	7.5494	h
Mixer 1|Valve Actuation Count Since	62	count

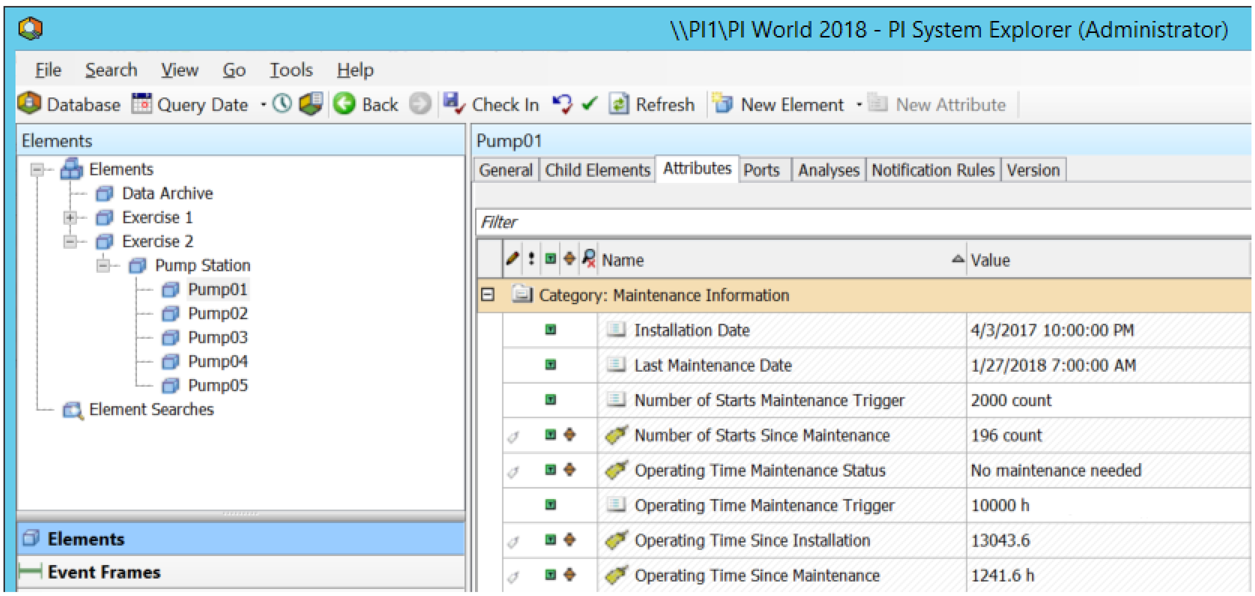
Name	Value	Units
Mixer 2|Last Maintenance Date	3/25/2018 12:00:00 AM	
Mixer 2|MTD Run Hours	55.889	h
Mixer 2|Previous Day Run Hours	16.317	h
Mixer 2|Run Hours Since Maintenanc	30.733	h
Mixer 2|Valve Actuation Count Since	34	count

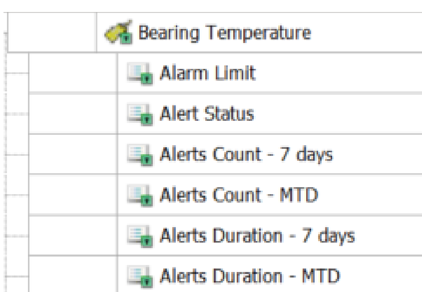
Name	Value	Units
Mixer 2|Last Maintenance Date	3/24/2018 12:00:00 AM	
Mixer 2|MTD Run Hours	7.4606	h
Mixer 2|Previous Day Run Hours	0	h
Mixer 2|Run Hours Since Maintenanc	7.4606	h
Mixer 2|Valve Actuation Count Since	53	count

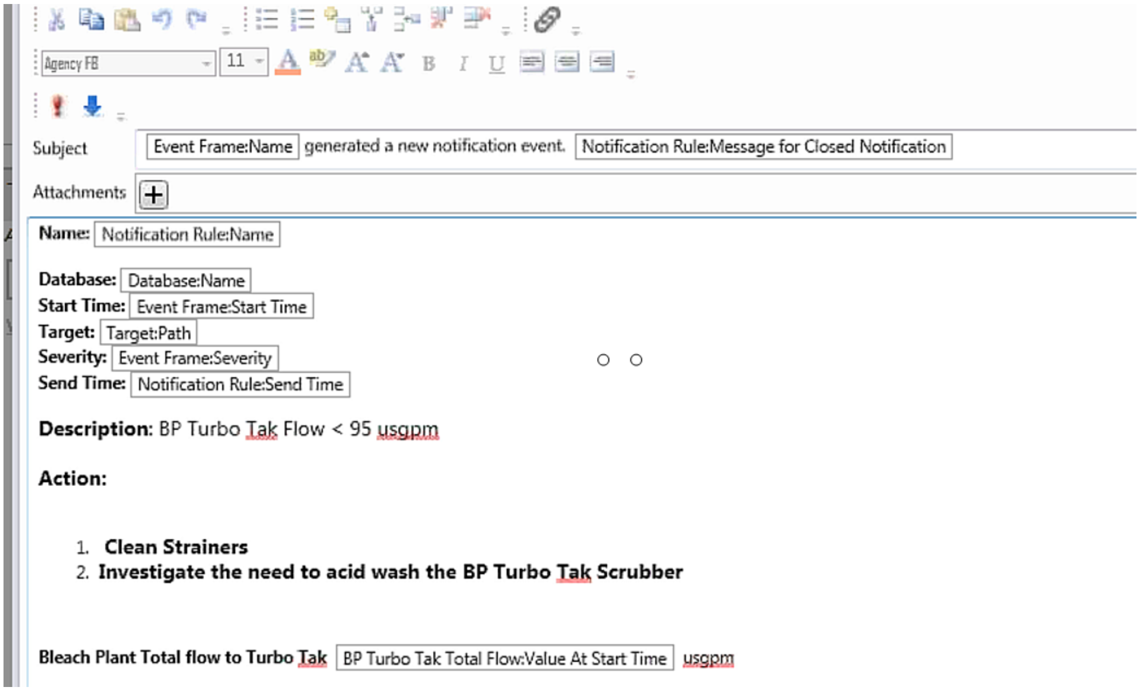
Exercise 2: Condition-based Maintenance (CbM)

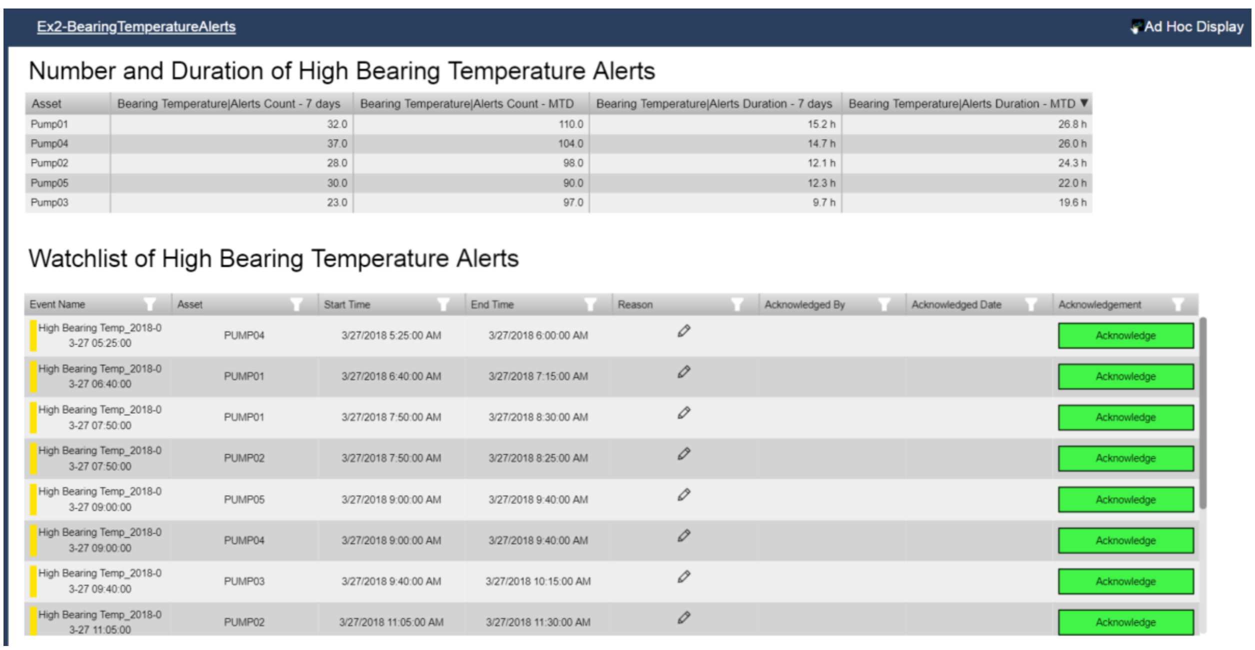
In this exercise, we assess the condition of an equipment by calculating metrics that can serve as leading indicators of equipment failure or loss of efficiency – for example, bearing temperature to understand the pump bearing condition.

We track the alerts for the bearing temperature and then discuss the use of PI Notification to send an email or use the web service delivery channel to notify a system (i.e. triggering a work order in a work management system such as SAP or IBM Maximo) for follow-up action. The bearing temperature events are viewed in a watchlist in PI Vision – see screens below.









Number and Duration of High Bearing Temperature Alerts

Asset	Bearing Temperature|Alerts Count - 7 days	Bearing Temperature|Alerts Count - MTD	Bearing Temperature|Alerts Duration - 7 days	Bearing Temperature|Alerts Duration - MTD ▼
Pump01	32.0	110.0	15.2 h	26.8 h
Pump04	37.0	104.0	14.7 h	26.0 h
Pump02	28.0	98.0	12.1 h	24.3 h
Pump05	30.0	90.0	12.3 h	22.0 h
Pump03	23.0	97.0	9.7 h	19.6 h

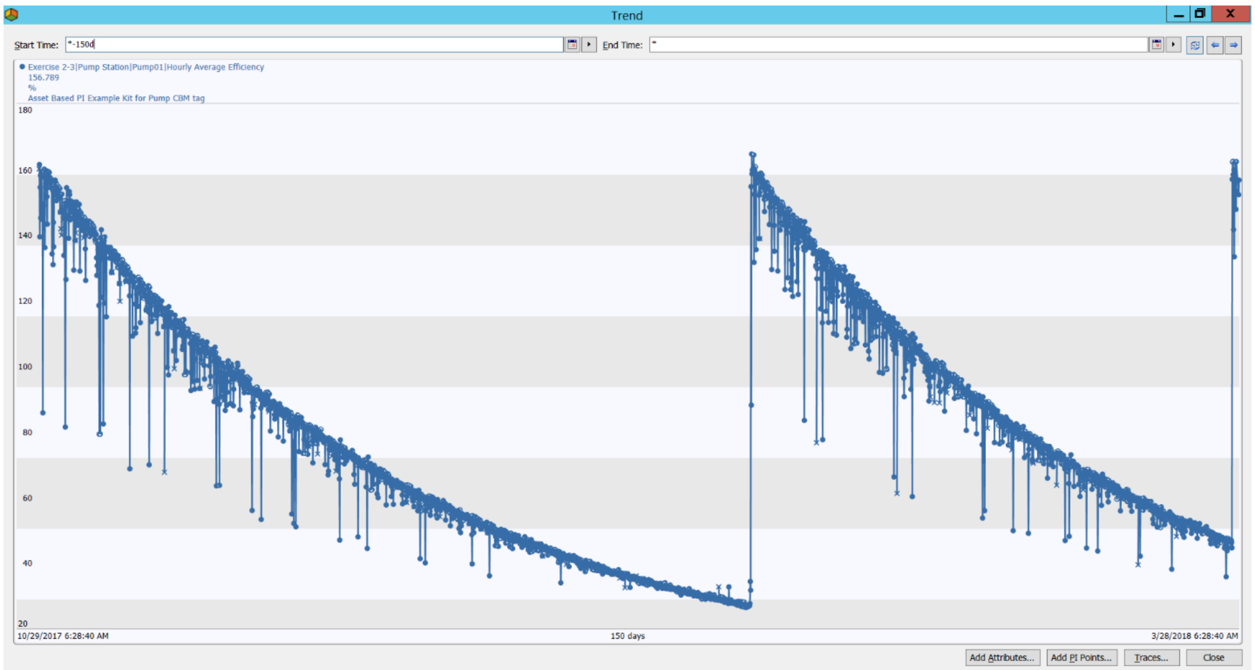
Watchlist of High Bearing Temperature Alerts

Event Name	Asset	Start Time	End Time	Reason	Acknowledged By	Acknowledged Date	Acknowledgement
High Bearing Temp_2018-03-27 05:25:00	PUMP04	3/27/2018 5:25:00 AM	3/27/2018 6:00:00 AM				Acknowledge
High Bearing Temp_2018-03-27 06:40:00	PUMP01	3/27/2018 6:40:00 AM	3/27/2018 7:15:00 AM				Acknowledge
High Bearing Temp_2018-03-27 07:50:00	PUMP01	3/27/2018 7:50:00 AM	3/27/2018 8:30:00 AM				Acknowledge
High Bearing Temp_2018-03-27 07:50:00	PUMP02	3/27/2018 7:50:00 AM	3/27/2018 8:25:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:00:00	PUMP05	3/27/2018 9:00:00 AM	3/27/2018 9:40:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:00:00	PUMP04	3/27/2018 9:00:00 AM	3/27/2018 9:40:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:40:00	PUMP03	3/27/2018 9:40:00 AM	3/27/2018 10:15:00 AM				Acknowledge
High Bearing Temp_2018-03-27 11:05:00	PUMP02	3/27/2018 11:05:00 AM	3/27/2018 11:30:00 AM				Acknowledge

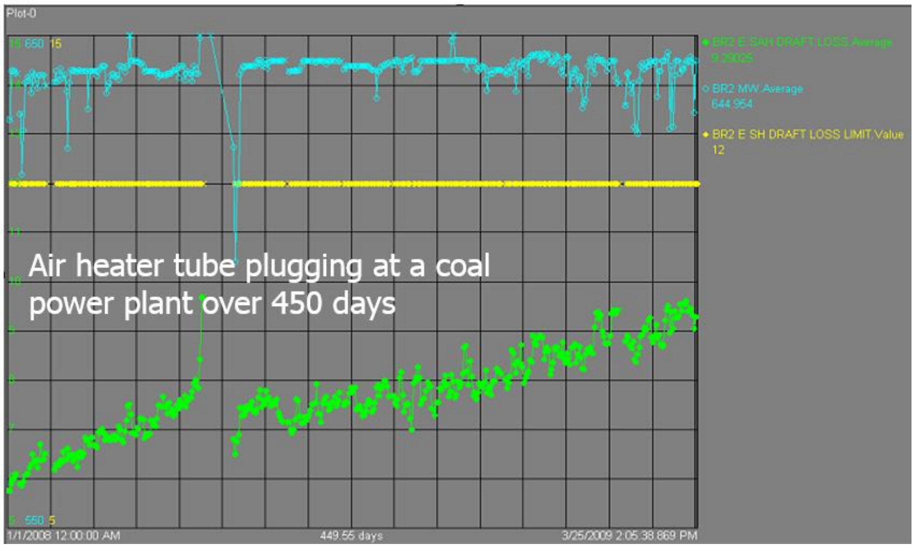
Exercise 3a: Predictive Maintenance (PdM) – Compressor efficiency

For certain classes of process equipment, their condition can be evaluated by monitoring some key metric, such as efficiency for a compressor, fouling for a heat-exchanger etc. Often, these metrics show a pattern with time – and, linear, piece-wise linear or non-linear trend can be extrapolated to estimate remaining-useful-life.

The screen below shows decreasing pump efficiency over time (100+ days). And, after maintenance, its efficiency is restored to a higher value. The trend can be extrapolated to schedule maintenance.



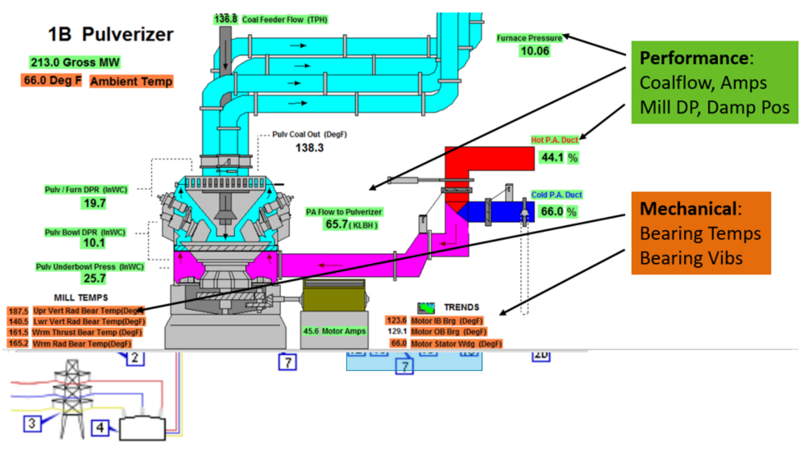
Another example from a coal power plant air heater is shown below. The green trace (with increasing Delta P) shows the heater getting increasingly plugged over a period of 450 days. The blue trace shows a nominal 650 MW production rate whenever the air heater Delta P is calculated. The yellow trace shows the maximum allowable DP i.e. 12 inches of H_2O.

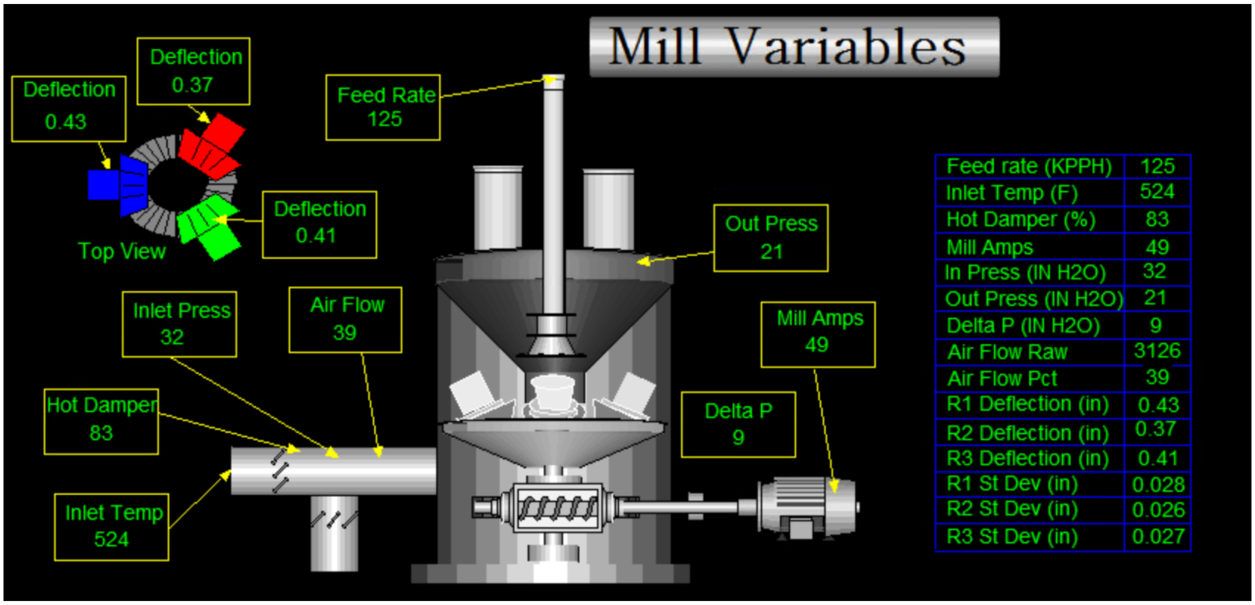


A linear extrapolation of the green trace will indicate that you have about 60-90 days before air heater DP reaches the maximum allowable limit and should be scheduled for maintenance.

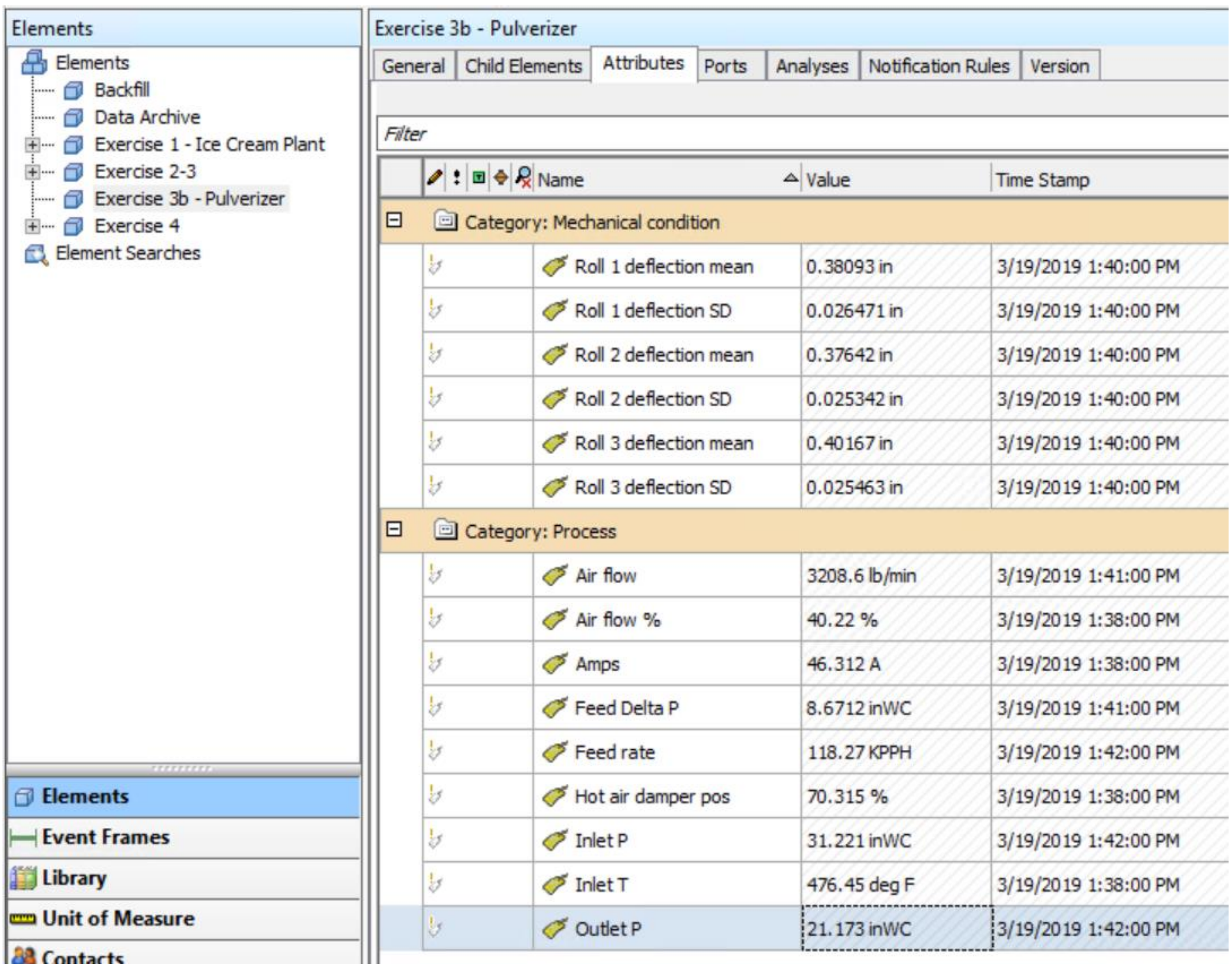
Exercise 3b: Predictive Maintenance (PdM) – Coal pulverizer bearing – early fault detection using APR (Advanced Pattern Recognition)

In this exercise, we use [ECG's](ECG's) APR (Advanced Pattern Recognition) based **Predict-It** to monitor a coal pulverizer in a power generation plant. APR uses similarity-based modeling to compare the pulverizer's current operation with its historical data and detects subtle changes in its run-time behavior to provide early warning fault detection.





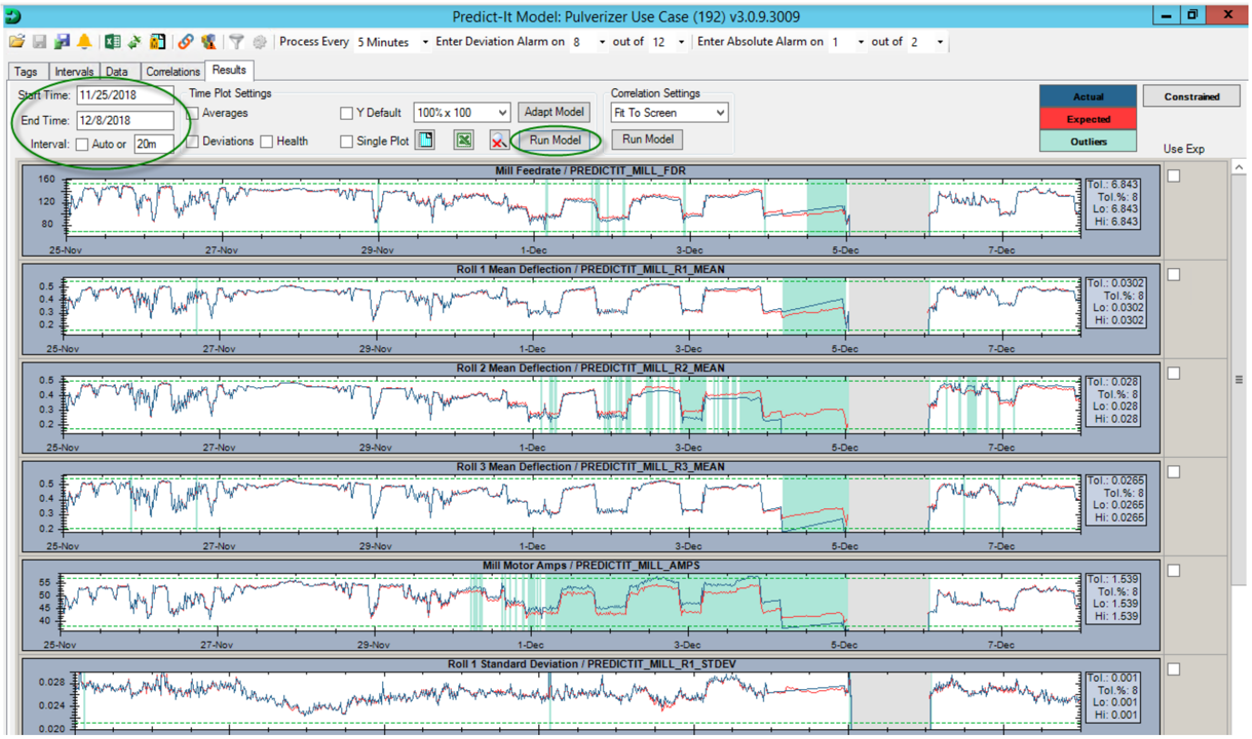
Mill Variables
Deflection 0.43
Deflection 0.37
Deflection 0.41
Top View
Feed Rate 125
Out Press 21
Inlet Press 32
Air Flow 39
Mill Amps 49
Hot Damper 83
Delta P 9
Inlet Temp 524
Feed rate (KPPH) 125
Inlet Temp (F) 524
Hot Damper (%) 83
Mill Amps 49
In Press (IN H2O) 32
Out Press (IN H2O) 21
Delta P (IN H2O) 9
Air Flow Raw 3126
Air Flow Pct 39
R1 Deflection (in) 0.43
R2 Deflection (in) 0.37
R3 Deflection (in) 0.41
R1 St Dev (in) 0.028
R2 St Dev (in) 0.026
R3 St Dev (in) 0.027



Elements
Elements
Backfill
Data Archive
Exercise 1 - Ice Cream Plant
Exercise 2-3
Exercise 3b - Pulverizer
Exercise 4
Element Searches
Elements
Event Frames
Library
Unit of Measure
Contacts
Exercise 3b - Pulverizer
General
Child Elements
Attributes
Ports
Analyses
Notification Rules
Version
Filter
Name
Value
Time Stamp
Category: Mechanical condition
Roll 1 deflection mean 0.38093 in 3/19/2019 1:40:00 PM
Roll 1 deflection SD 0.026471 in 3/19/2019 1:40:00 PM
Roll 2 deflection mean 0.37642 in 3/19/2019 1:40:00 PM
Roll 2 deflection SD 0.025342 in 3/19/2019 1:40:00 PM
Roll 3 deflection mean 0.40167 in 3/19/2019 1:40:00 PM
Roll 3 deflection SD 0.025463 in 3/19/2019 1:40:00 PM
Category: Process
Air flow 3208.6 lb/min 3/19/2019 1:41:00 PM
Air flow % 40.22 % 3/19/2019 1:38:00 PM
Amps 46.312 A 3/19/2019 1:38:00 PM
Feed Delta P 8.6712 inWC 3/19/2019 1:41:00 PM
Feed rate 118.27 KPPH 3/19/2019 1:42:00 PM
Hot air damper pos 70.315 % 3/19/2019 1:38:00 PM
Inlet P 31.221 inWC 3/19/2019 1:42:00 PM
Inlet T 476.45 deg F 3/19/2019 1:38:00 PM
Outlet P 21.173 inWC 3/19/2019 1:42:00 PM

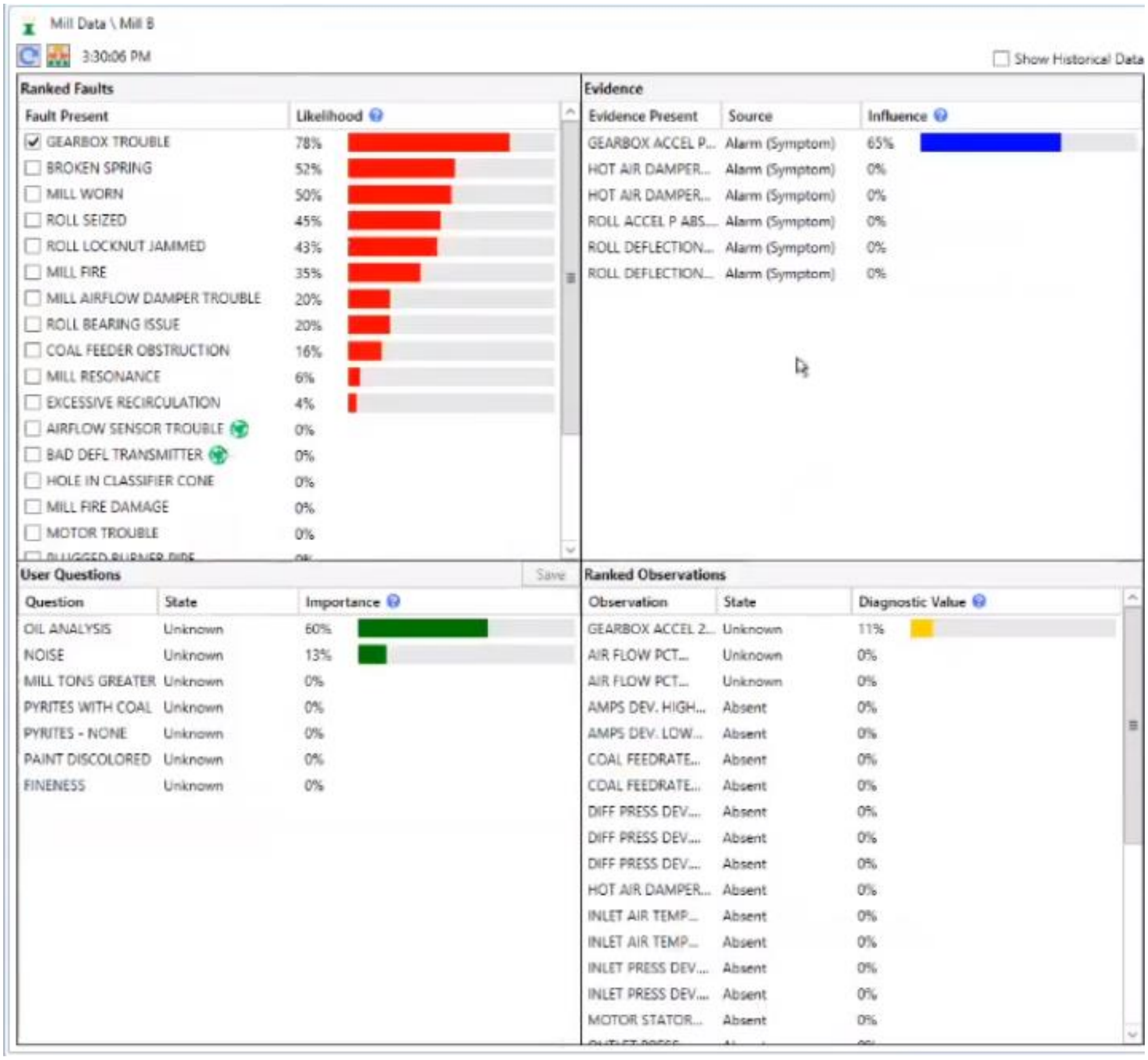
The Exercise includes creating an APR model for the pulverizer – i.e. selecting the relevant sensor data (coal feeder rate, pulverizer motor amps, pressures, roll deflection etc.), training the model using historical data with "good operations," validating the model with previously unseen (by the model) data and finally deploying the model to run in real-time.

Screen below shows the pulverizer **Mill Motor Amps** indicating a fault status (green shaded) starting on 01-Dec, as much as 5 days prior to its final failure on 05-Dec.



The APR model, optionally, can write the expected values for the pulverizer back to PI; this allows you to use it with AF Analytics, PI Notification, PI Vision etc.

The lab will also include a walk-through regarding such early fault detection as part of a fault-tree in terms of symptoms and likelihood of failure.



The lab will also include a brief walk through of other predictive analytics use cases such as:

- Predicting remaining useful life (RUL) based on the history of engine operations data and its failures - more
- Anomaly detection in an HVAC air-handler - more

Exercise 4: Multiple condition assessment rules and asset health score

In this Exercise, you apply the appropriate condition assessment rules and corresponding weighting factors to process/equipment measurements to calculate an overall asset health score.

It uses AF Analytics to convert a “Raw Value” (sensor data) to a normalized i.e. a “Case Value”. And then, by applying a Weight%, it is transformed to a Score.

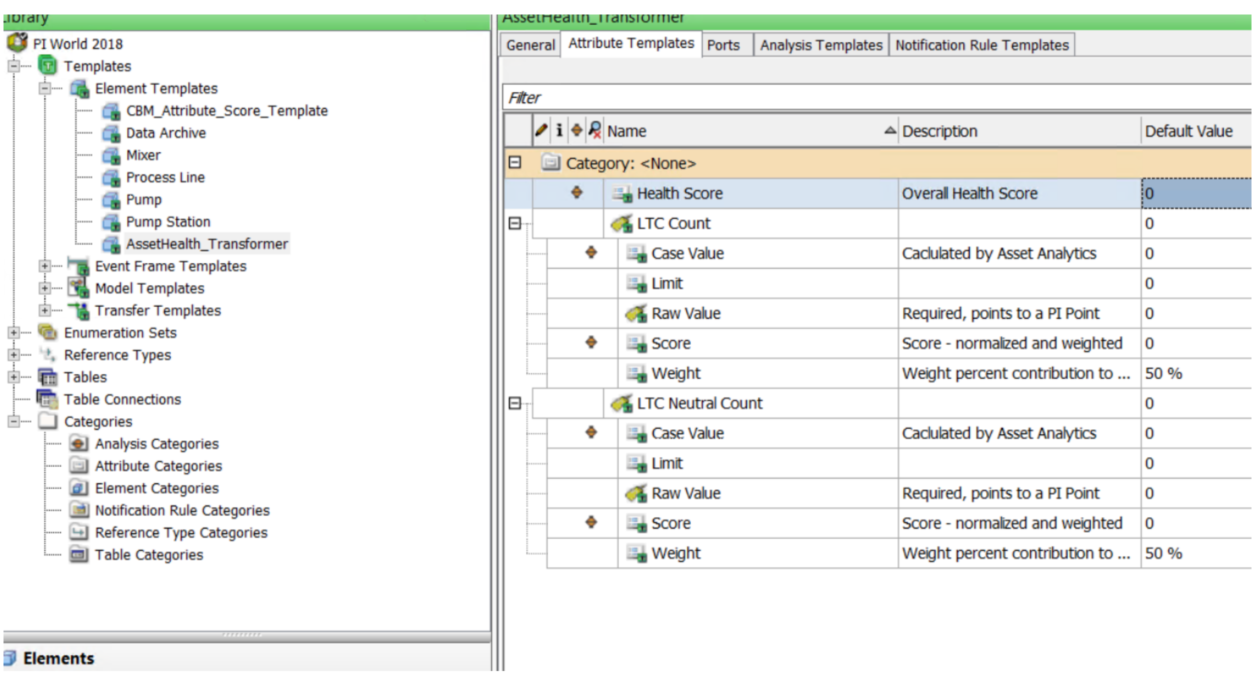
Each measurement gets a normalized weighted score (0 to 10) by applying a condition assessment rule. And, then the normalized scores are rolled up to arrive at a composite asset health score. The Weight% applied to each attribute depends on its contribution to the overall asset health.

The composite asset health score ranges from 0 to 10 (0=Good, 10=Bad)

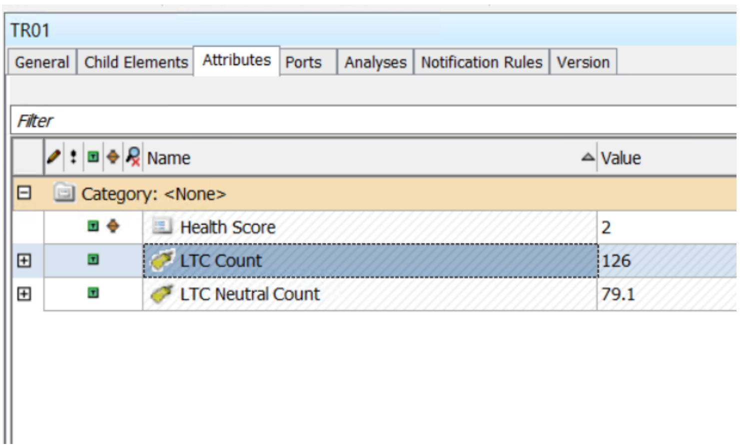
A Transformer asset health score example is used with the following measurements:

- LTC counter operations (LTC= Load Tap Changer)
- LTC through neutral count
- DGA (dissolved gas analysis) detectable acetylene
- DGA High gas rate of change
- Low dielectric
- High water
- Low nitrogen pressure

An example Transformer template is as below:



And, as you configure Transformers using these templates, the composite health score is periodically calculated by PI System Asset Analytics.



The composite health score for transformer TR01 is 2 i.e. asset is in good health (0=Good, 10=Bad).

PI System software

The VM (virtual machine) used for this lab has the following PI System software installed:

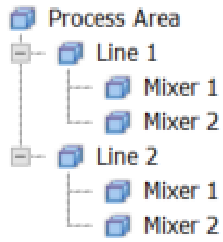
Software	Version
PI Data Archive	2017 R2
PI Asset Framework (PI AF) server	2018
PI Asset Framework (PI AF) client (PI System Explorer)	2018
PI Analysis & PI Notifications Services	2018
PI Vision	2017 R2
PI Web API	2017 R2

For details on PI System software, please refer to:
http://www.osisoft.com/pi-system/pi-capabilities/product-list/

Exercise 1 – Usage-based Maintenance (UbM)

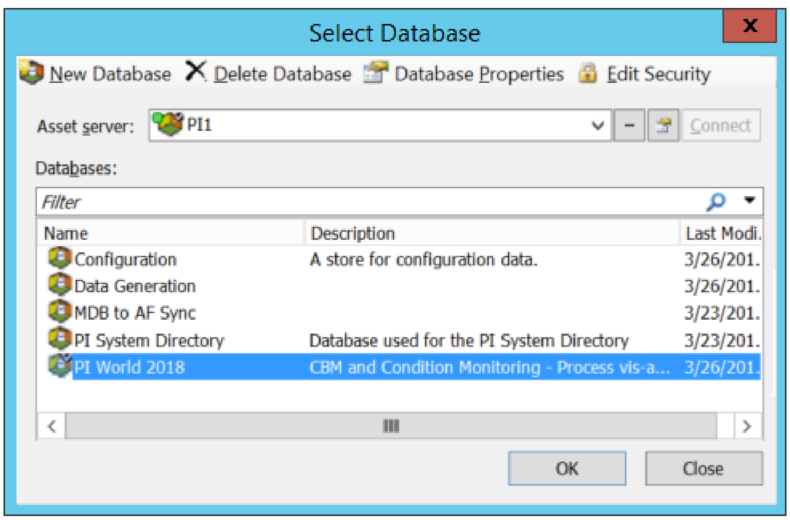
In this exercise, equipment run-hours and valve actuation counts are calculated to serve as a basis for usage-based maintenance.

There are two process lines in an ice-cream factory – Line 1 and Line 2, with two mixers on each line.

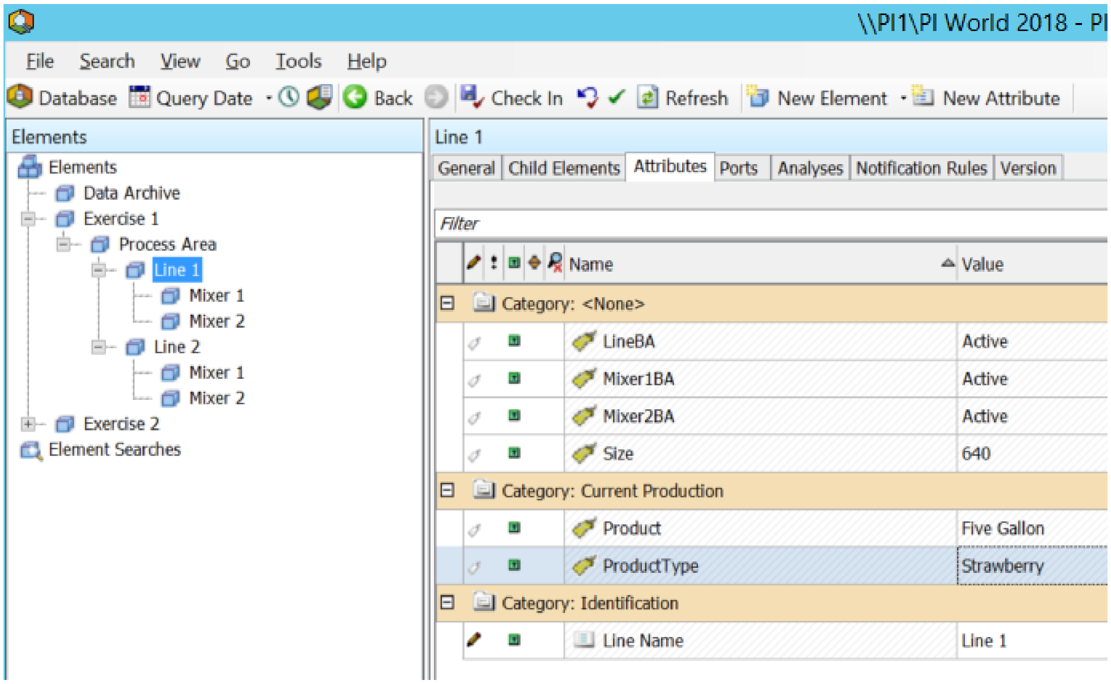


Step 1: Exploring the Asset Framework structure

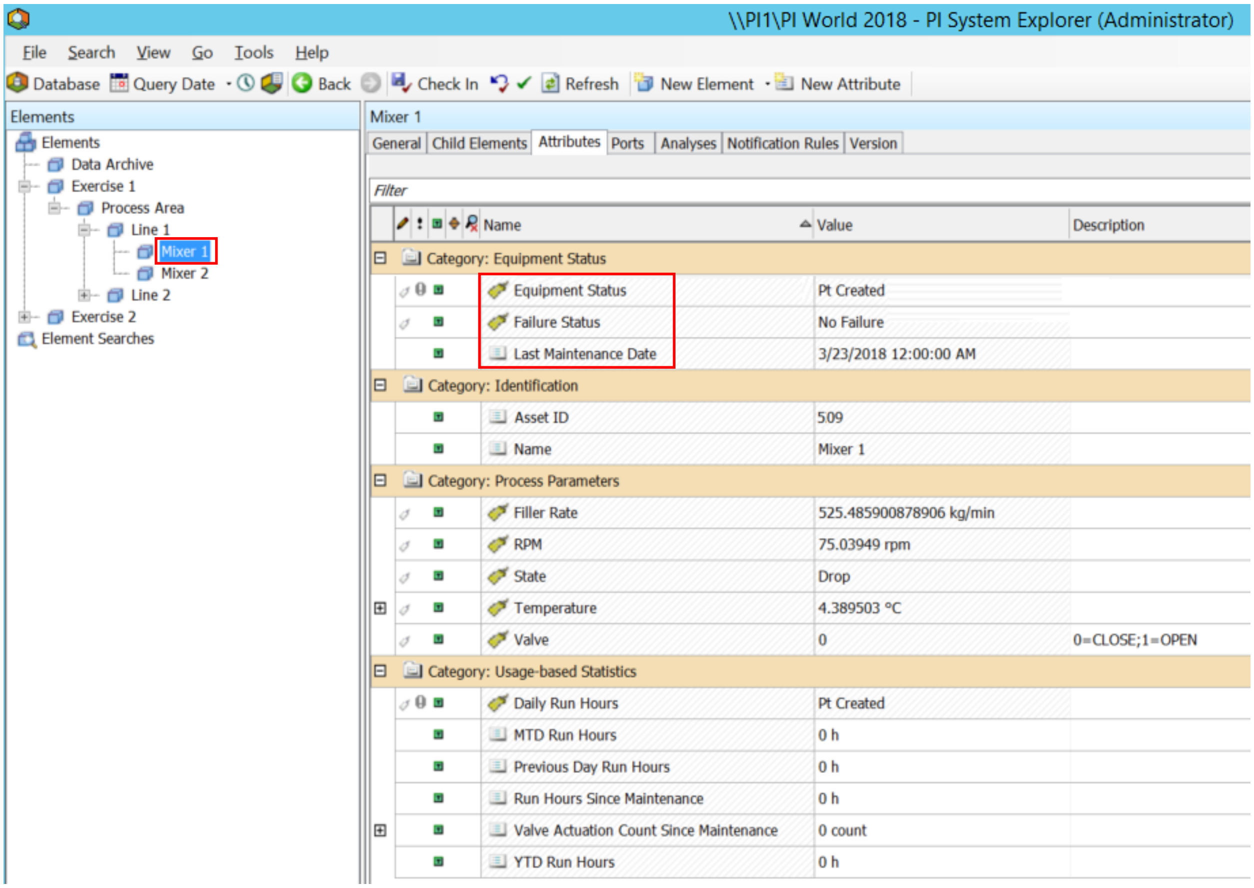
1. Open **PI System Explorer**; connect to the **PI World 2018** AF database.
   If the top bar of the PI System Explorer window does not already show **\\PI1\PI World 2018**, then click on the top toolbar Database button to select the PI AF database named **PI World 2018**.



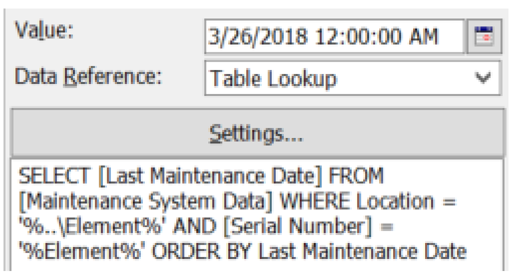
2. From the **Elements** section, navigate to **Exercise 1 > Process Area > Line 1**.
   The **Attributes** tab shows the current production on Line1.



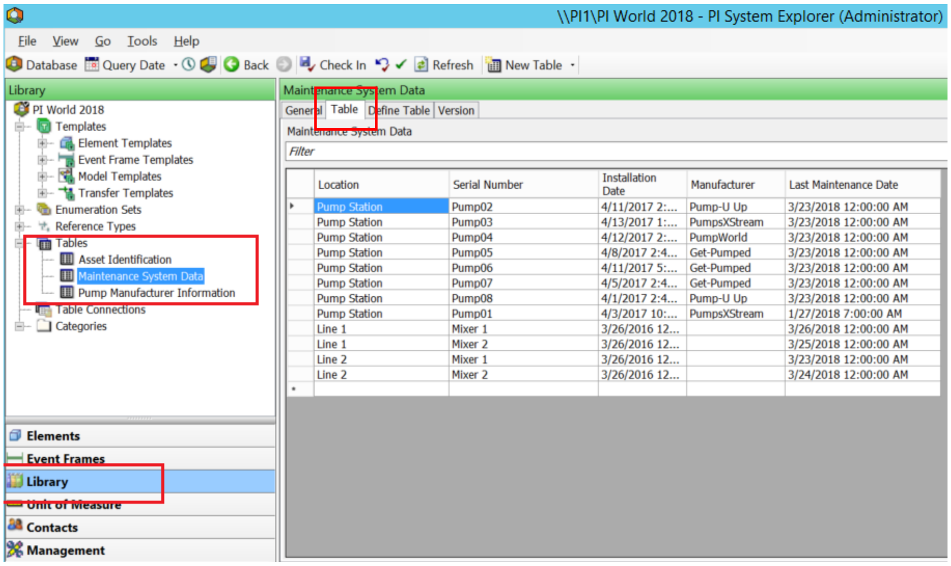
3. Drill-down under Line 1 > **Mixer 1**, then click on the **Attributes** tab for Mixer 1.



a. Note the **Last Maintenance Date** attribute.
   Review the attribute configuration (in the right-side panel); SELECT statement retrieves the date from a table.



b. To view the full table, go to the **Library** section of PI System Explorer, under **Tables > Maintenance System Data** section.
   Select the **Table** tab to visualize the data.
   Typically, this table queries an external system such as your maintenance database and is refreshed with new values on a periodic basis.



The ice-cream mixers each use a motor; its RPM (revolutions per minute) value is measured.

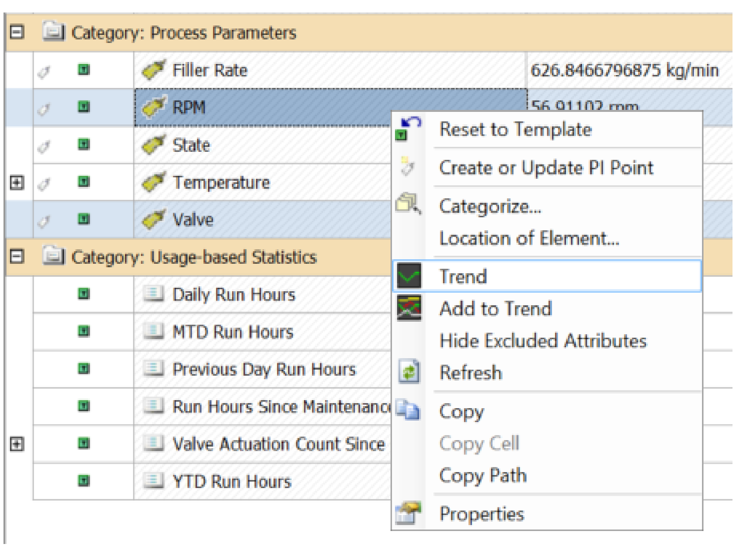
A valve is opened to allow the product to flow in and be mixed. We would like to calculate the running hours for the mixers but there is no direct measurement to indicate its running/idle status of the mixer.

Step 2: Creating a Status attribute

The mixer is inferred to be "in production" when it is running and the valve is open.

Let's create an analysis to store the equipment status. We also know that the equipment provides a failure indicator which can be useful for OEE type calculations.

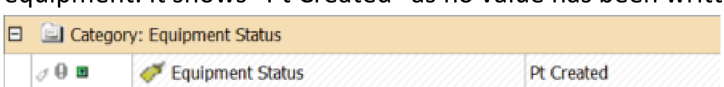
1. Go to the **Elements** section of PI System Explorer, under our Line 1 > **Mixer 1** equipment. Look at the **RPM** and **Valve** attributes.
   Select both attributes using the Ctrl key on your keyboard, and then right-click one of the two attributes to select **Trend**.



Note the RPM in blue and Valve status in red.

The equipment is "in production" when both the valve is open (value=1) and the RPM has a positive value (value>0).

2. Note **Equipment Status** attribute that we have created as a placeholder for the status of the equipment. It shows "Pt Created" as no value has been written to it yet.

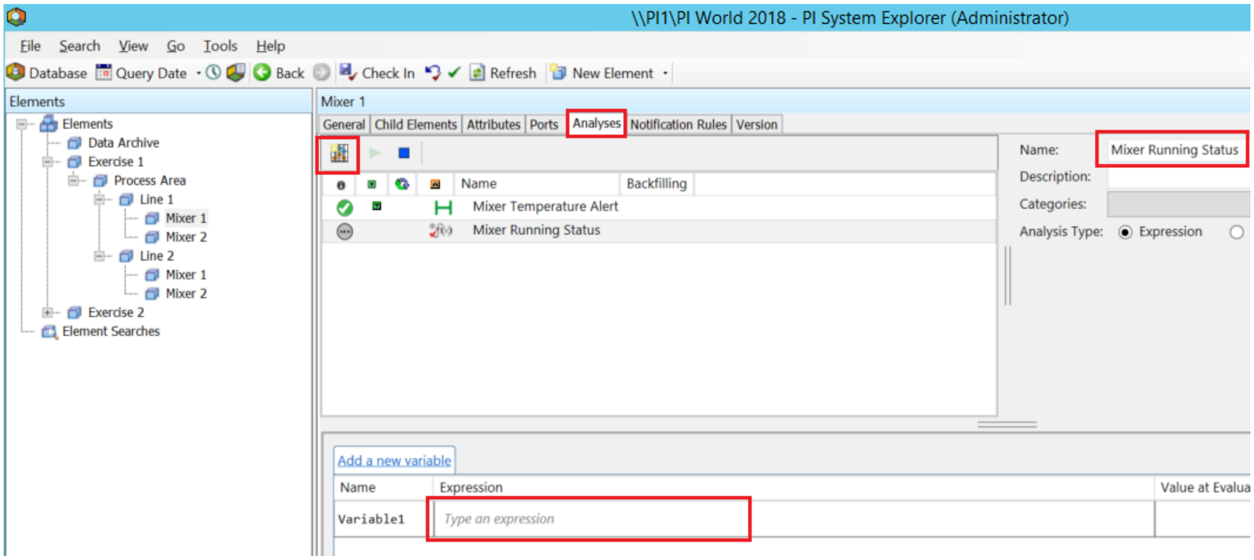


3. This Equipment Status attribute uses an enumeration set called **Equipment Status**, where:

Value	Name
0	Idle
1	Running
2	Failure

The Enumeration set is available from Library > Enumeration Sets > Equipment Status.

4. Switch to the **Analyses** tab for Mixer 1 and create a new Expression analysis called **Mixer Running Status**.



4. To configure this expression, we use the RPM attribute and the Failure Status attribute.
    a. In the Variable1 expression field, enter the following expression; you can use the "intellisense" like suggestions to speed up the process of writing this expression.

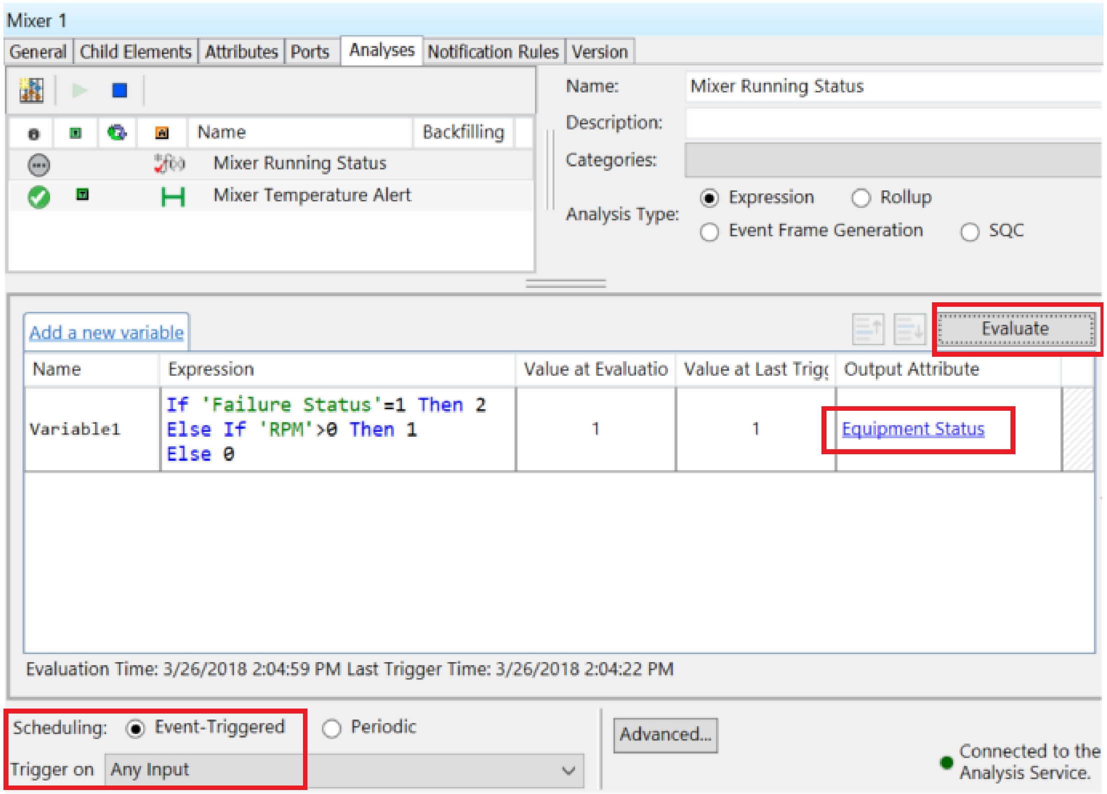
Name	Expression
Variable1	If 'Failure Status'=1 Then 2 Else If 'RPM'>0 Then 1 Else 0

Note: Use Shift+Enter to start a new line while writing an Analysis Expression. Use // in front of a line to mark it as a comment.

Note: In the case of OEE calculation, we may consider the whole equipment to be "producing/running" only when the Valve is open and the motor is running, and the expression for the equipment status will be:

Name	Expression
Variable1	If 'Failure Status'=1 Then 2 Else If ('RPM'>0 And 'Valve'=1) Then 1 Else 0

    b. Map the expression result to the **Equipment Status** attribute (see next picture).
    c. Leave the scheduling option to **Event-Triggered** and **Any Input**.
    d. Click the **Evaluate** button to check the current value.

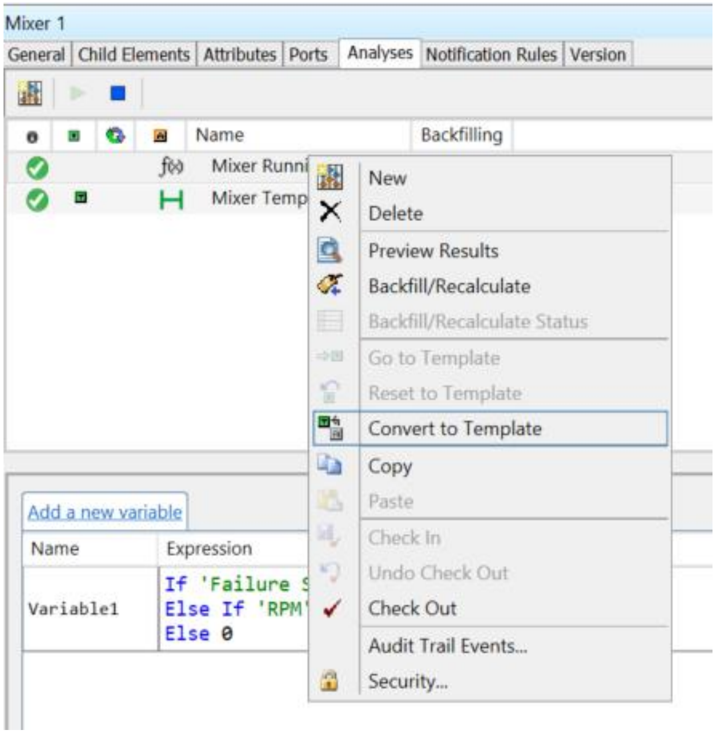


e. Click the Check In button in the top toolbar to save the changes. The analysis will attempt to start and show Running (green checkmark) if no error is present.

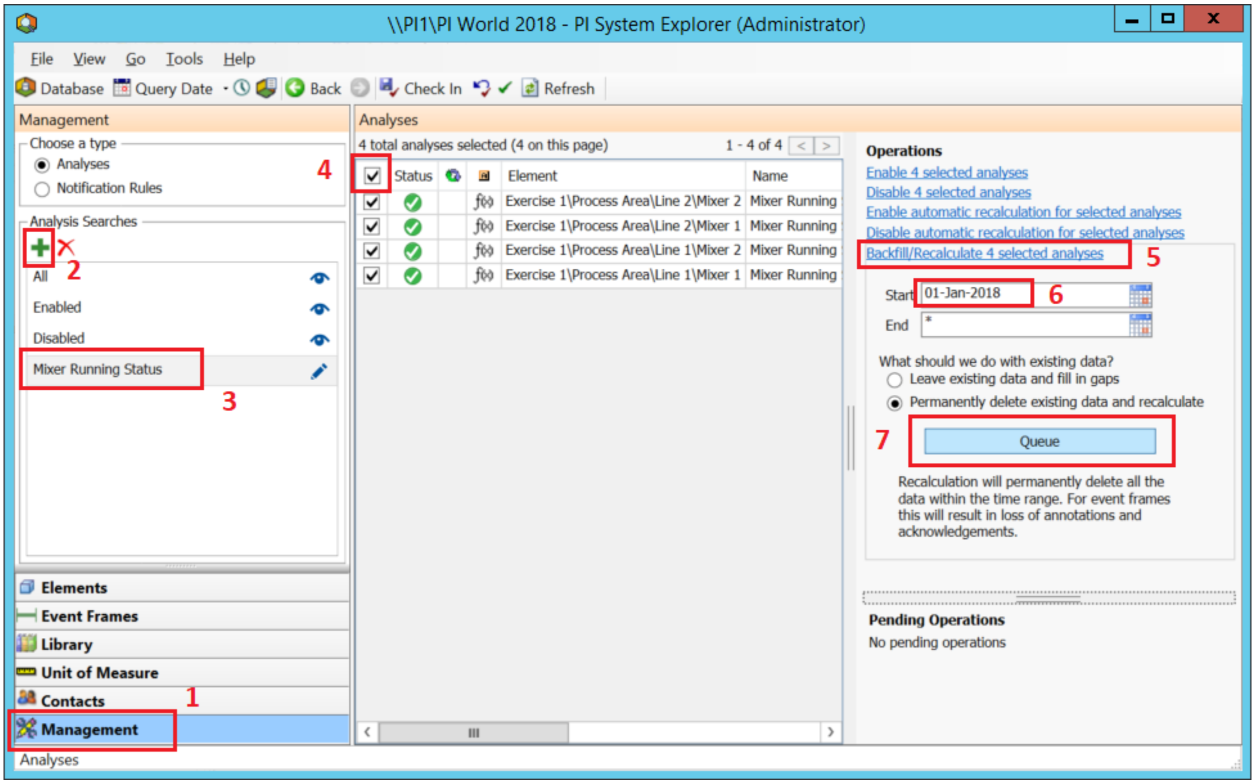
5. The expression is currently only available to Mixer 1.
   To enable it for all other mixers, right-click the **Mixer Running Status** analysis and select **Convert to Template**.

   Next, click on Check In to save the changes.

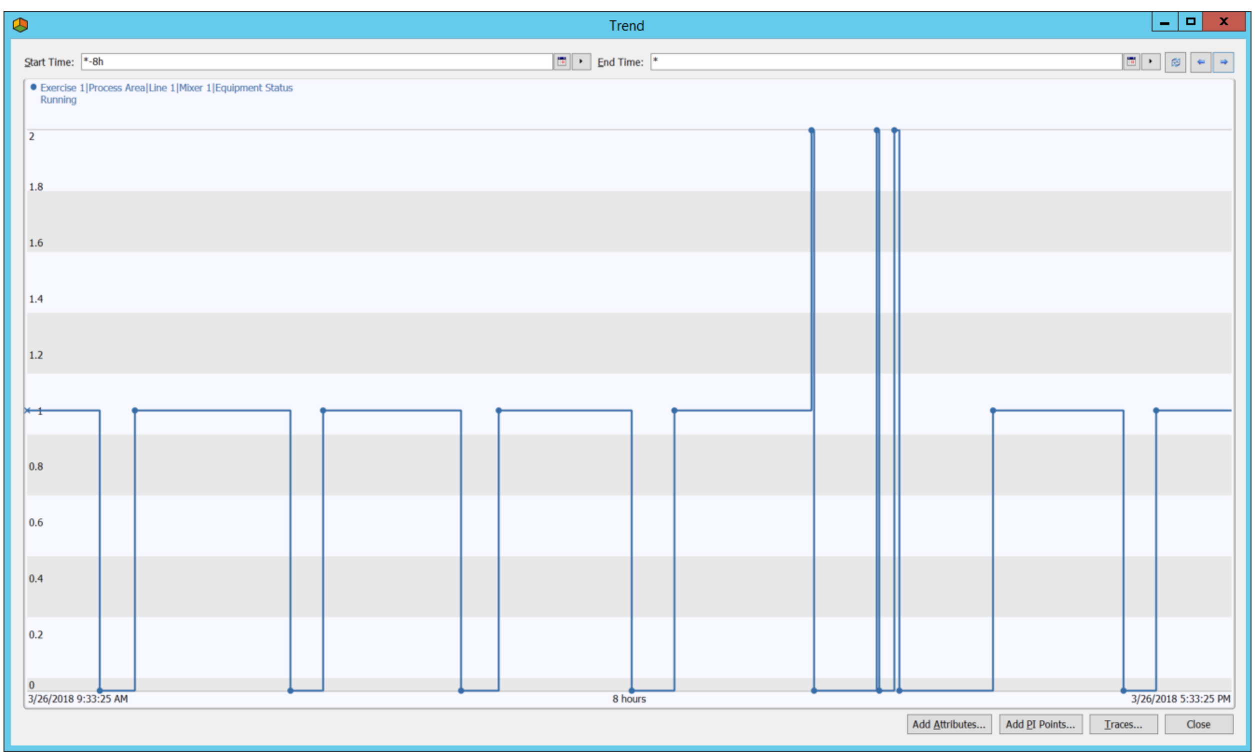
   The calculations can now be applied to all four mixers.



6. We will now backfill the status attribute since the beginning of the year (since 01-Jan-2018). From the **Management** section of PI System Explorer, select (see below picture for guidance):
    1. Management
    2. The **Plus** sign to add an analysis search
    3. Enter a name like **Mixer Running Status** and add a search criterion where the **Name = Mixer Running Status** (or the name you gave the analysis created in the previous step). Once created, make sure to select it to filter the list of retrieved analyses in the center.
    4. Check the checkbox to select the four (4) Mixer Running Status analyses.
    5. Click the **Backfill/Recalculate 4 selected analyses** link.
    6. Enter a start date of **01-Jan-2018** and leave the end date to * (right now).
    7. Click the **Queue** button to start the backfilling.



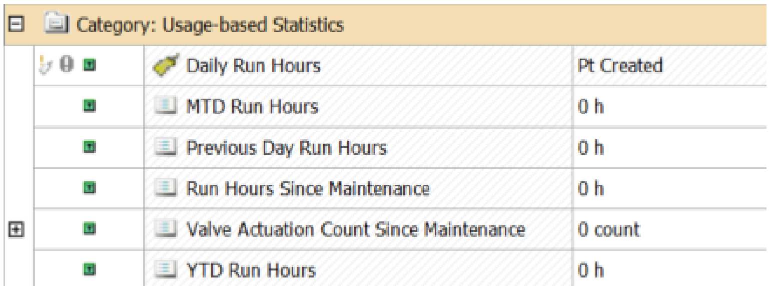
7. Confirm that the backfill was successful; in the **Elements** section, look at the **Equipment Status** attribute of Line 1 > Mixer 1 (0=Idle, 1=Running, 2=Failure).
   Right-click on the attribute to select **Trend**. View the equipment status for the last 8 hours, from ***-8h** to *****.



Step 3: Creating usage-based run hours attributes

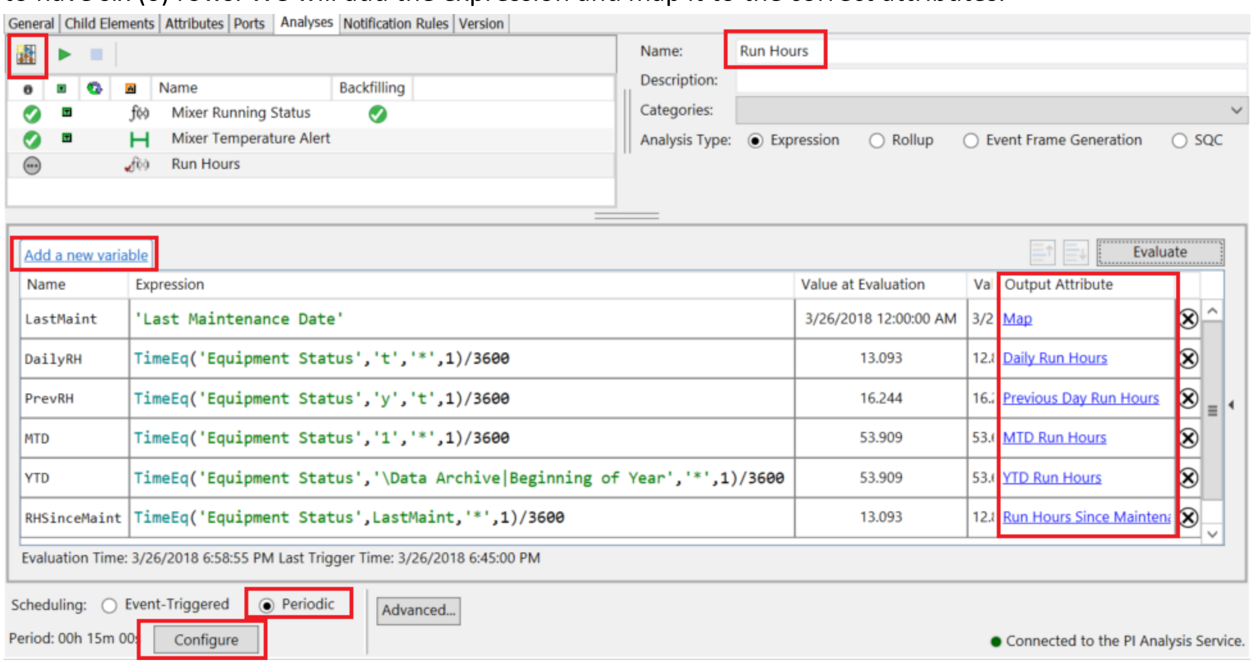
Now that we have an equipment status attribute, we can use it to create usage-based counters for different time ranges (daily, previous day, MTD, YTD, etc.).

1. From the Elements section, select a Mixer and look at its attributes under the **Usage-based Statistics** category.



   We will first populate the run-hours attributes with an analysis, then the valve actuation count with a second analysis.

2. For the run-hours attributes, navigate to the **Analyses** tab for Exercise 1 > Line 1 > Mixer 1 and create a new Expression analysis named **Run Hours**.
3. Use the **Add a new variable** link above the expressions section to add multiple rows/expressions to have six (6) rows. We will add the expression and map it to the correct attributes.



   The TimeEq() function calculates the total time in seconds, within a range, that an attribute value is equal to a specified value. In our case, we are looking for the time where the Equipment Status attribute was equal to 1 (where 1=Running).

Note: In Expressions, PI Time Format can be used to specify relative times. For instance, '*' means now, 't' means today at midnight, 'y' means yesterday at midnight, and '1' means the first day of the current month at midnight.

Name	Expression	Description
LastMaint	'Last Maintenance Date'	Last maintenance date.
DailyRH	TimeEq('Equipment Status', 't', '*',1)/3600	Today's run hours.
PrevRH	TimeEq('Equipment Status', 'y', 't',1)/3600	Yesterday's run hours.
MTD	TimeEq('Equipment Status', '1', '*',1)/3600	Month-to-date run hours, '1' meaning the first day of the current month.
YTD	TimeEq('Equipment Status', '\Data Archive|Beginning of Year', '*',1)/3600	Year-to-date run hours, with an attribute of the root *Data Archive* element used to store the first day of the year, names *Beginning of Year*.
RHSinceMaint	TimeEq('Equipment Status', 'Last Maintenance Date', '*',1)/3600	Run hours since the last maintenance date attribute value.

4. Configure the scheduling to be **Periodic**, every 5 minutes. This is probably too often for a production environment, but for the purpose of the lab, we want the results to be written quickly.

Note: These calculations, especially Year-To-Date (YTD) can be very expensive on the Analysis service. It is recommended to execute them periodically to ensure you have control on how often they are being triggered.

A more efficient way is to create a Run Hours totalizer analysis that would increment the lifetime or YTD run hours every day at midnight, and where the expression will be:

PreviousDayLifetimeTotal + DailyRunHours = NewLifetimeTotal

5. Click the **Evaluate** button to ensure no error is present. Next, click the top-toolbar Check In button to save the changes and start the analysis.
6. Right-click the **Run Hours** analysis and select **Convert to Template** to create it for the other mixers as well. Again, click on the Check In button to commit the changes.
7. Go back to the **Attributes** tab and click the **Refresh** button from the top toolbar. The attributes should show the correct run hours value.

Step 4: Creating a usage-based counter (valve actuation)

A valve, like other pieces of equipment, can wear with usage by actuation.

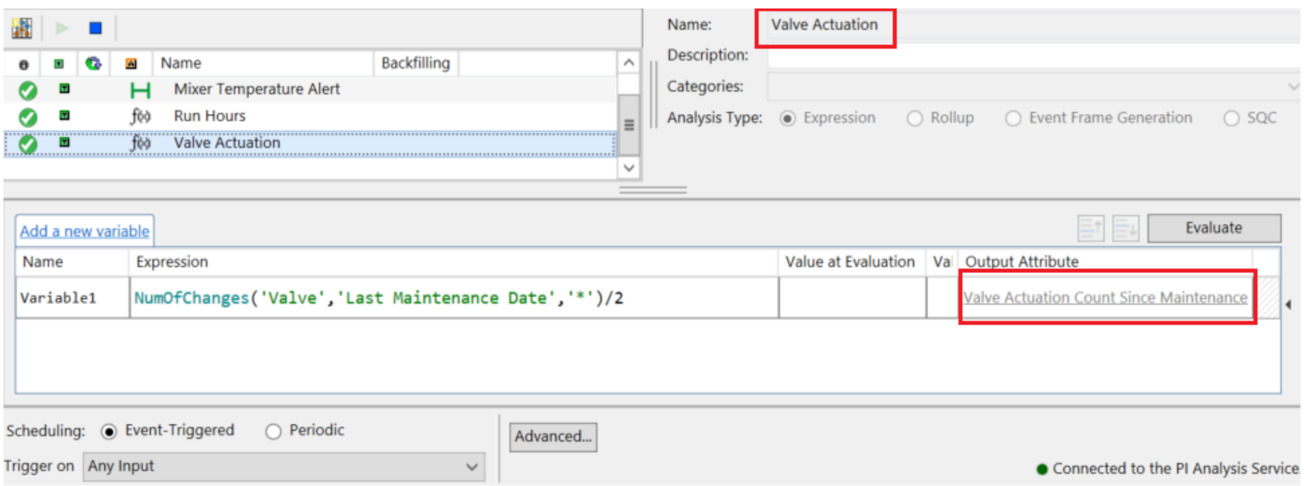
The valve on each mixer has a status shown in the **Valve** attribute, Close=0 and 1=Open.

We will create an analysis to count the number of 0-1 occurrences since the last maintenance date. As in the previous step, this calculation can be performed on different time ranges. Note that we will assume the same maintenance date for the whole asset, although you could have different maintenance dates for different components of an asset.

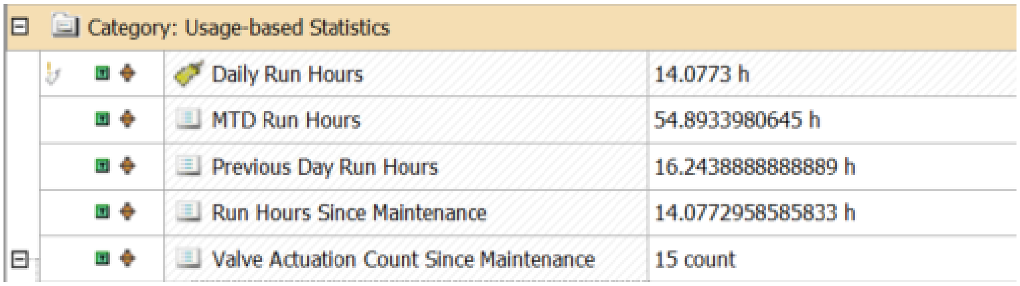
1. Under the **Analysis** tab, create a new expression analysis, named **Valve Actuation**.
2. Use the **NumOfChanges()** expression to calculate the number of actuations that occurred on the valve since the last maintenance date. The NumOfChanges() function returns the number of changes in value for an attribute within a specified time range. It will not consider equal consecutive values as a change (for example if a PI Tag was receiving and recording consecutive 1 values as in 0-1-1-1-1-1-0-1-0, this would count for 4 changes). We are dividing by 2 as NumOfChanges() will count the 0-1 and the 1-0 as changes.

Name	Expression
Variable1	NumOfChanges('Valve', 'Last Maintenance Date', '*')/2

3. Map the expression to the **Valve Actuation Count Since Maintenance** attribute.
4. Leave the scheduling to **Event-Triggered** and **Any Input**. The calculation result may come in a little while since it will be performed only when the Valve attribute will get a new value.



5. Evaluate the expression and if successful, click the Check In button to save the changes and start the analysis.
6. Right-click the **Valve Actuation** analysis to select **Convert to Template**. Click a second time on the Check In button to commit the changes.
7. Check the result from the **Attributes** tab (again it may take a while before the value shows up, use the **Refresh** button from the top toolbar to force a refresh of the attributes value).



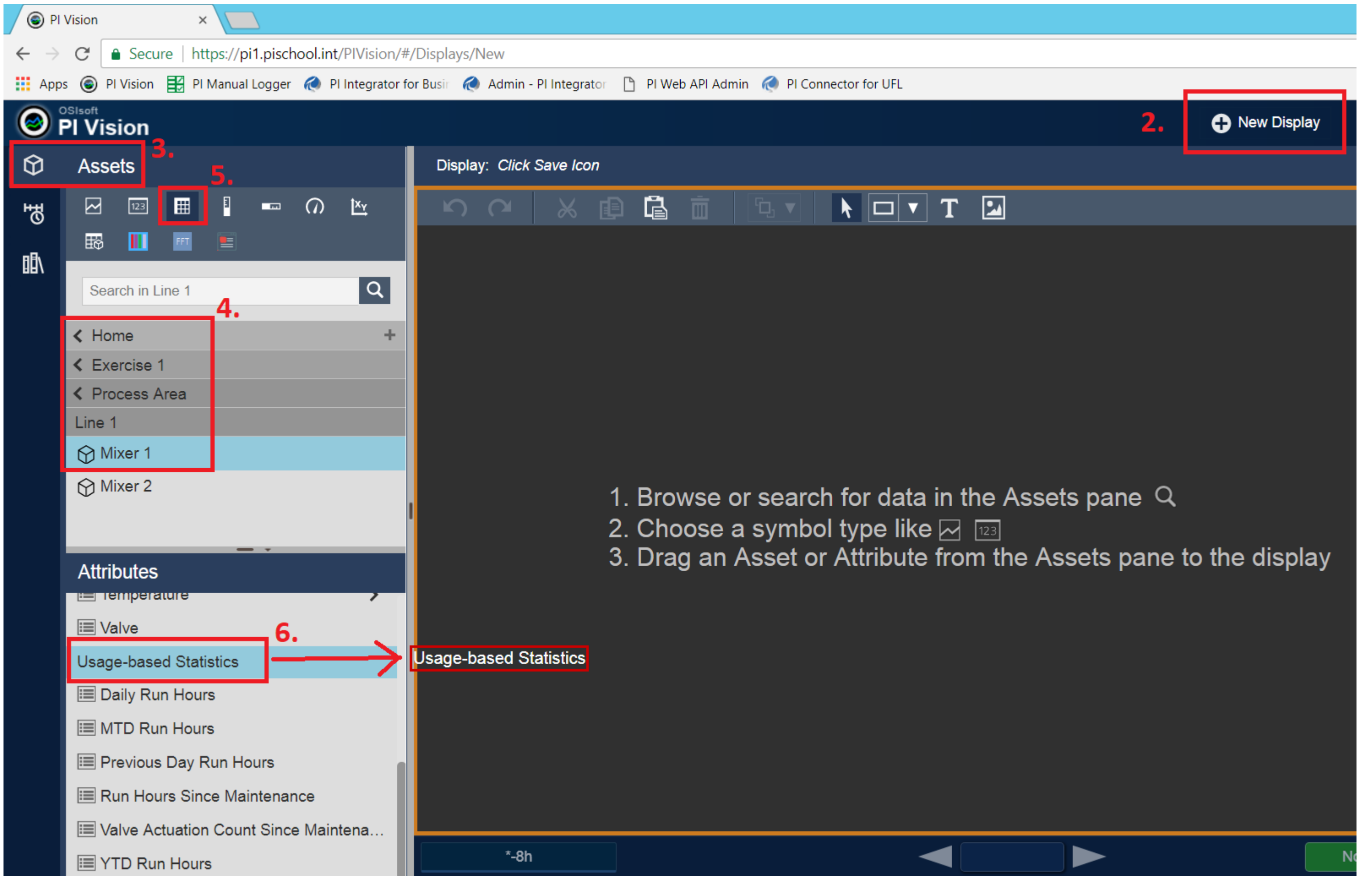
Category: Usage-based Statistics	
Daily Run Hours	14.0773 h
MTD Run Hours	54.8933980645 h
Previous Day Run Hours	16.2438888888889 h
Run Hours Since Maintenance	14.0772958585833 h
Valve Actuation Count Since Maintenance	15 count

Note: Similarly, a counter for the number of failures on the equipment can be created and used towards usage-based maintenance trigger. This will be explored as part of Exercise 2.

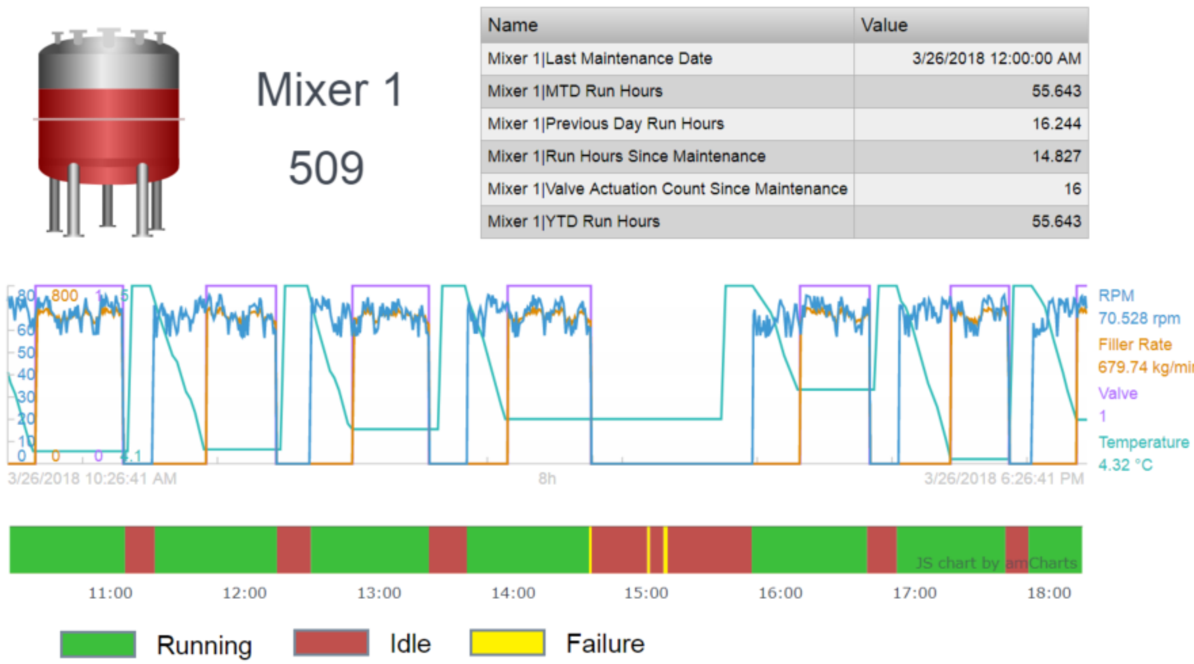
Step 5: (Optional) Comparing equipment on their usage-based statistics

We will use PI Vision to compare the status and the run hours and valve actuation attributes for the four mixers.

1. Open **Google Chrome** from the taskbar shortcut and then click on the **PI Vision** shortcut from the Bookmarks bar (this will bring you to https://pi1.pischool.int/PIVision).
2. You can explore the displays available, but for this step, click the **+ New Display** button in the top upper right.
3. Select the **Assets** pane on the left side.
4. Drill-down (click on the ">") to **PI World 2018 > Exercise 1 > Process Area > Line 1 > Mixer 1**.



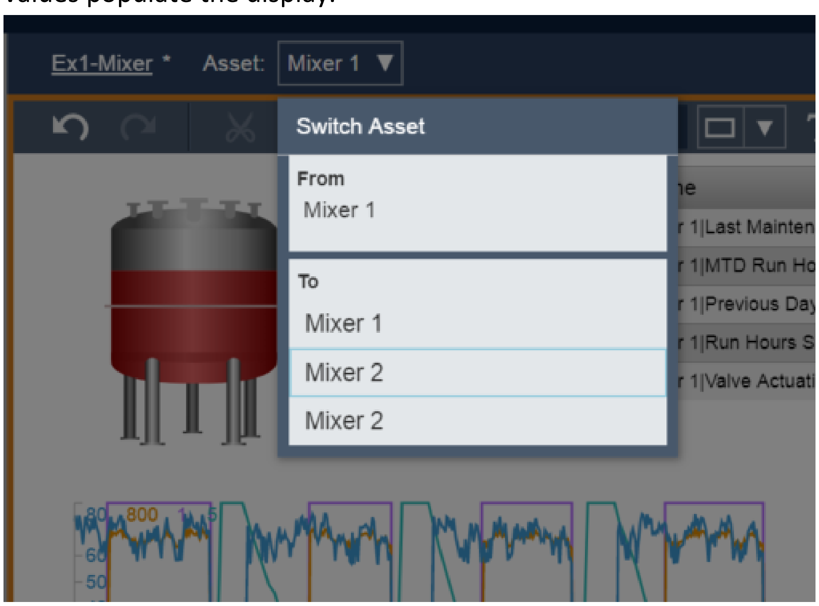
5. Select the **Table** symbol ( ) at the top.
6. Drag and drop the Usage-based Statistics category on the display. You can right-click the table that was added and select **Configure Table**, to keep only the **Name**, **Value**, and **Units** columns, and customize the formatting of the display.
7. You can then add the **Process Parameters** with a **Trend** symbol ( ), and also use the custom **String Values Plot** symbol ( ) to show the **Equipment Status** attribute.
8. The end result could look similar to (note the tank symbol is coming from the **Graphic Library** left-side pane ( ), under the **Tanks** section):



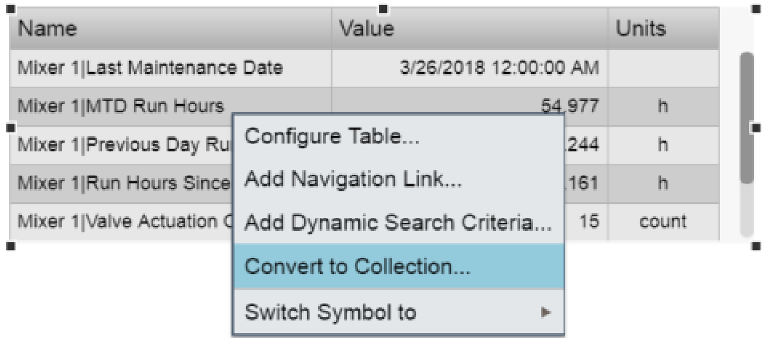
Name	Value
Mixer 1|Last Maintenance Date	3/26/2018 12:00:00 AM
Mixer 1|MTD Run Hours	55.643
Mixer 1|Previous Day Run Hours	16.244
Mixer 1|Run Hours Since Maintenance	14.827
Mixer 1|Valve Actuation Count Since Maintenance	16
Mixer 1|YTD Run Hours	55.643

Note: The **String Values Plot** custom PI Vision symbol is not officially supported. However, this symbol and several other useful symbols provide value and can be downloaded from GitHub: (https://github.com/osisoft/PI-Vision-Custom-Symbols).

9. Once completed, use the **Switch Asset** drop-down to select another Mixer and see the new values populate the display.



10. It is possible to have the statistics for all four mixers side by side. Simply right-click the Table symbol and select **Convert to Collection…**



11. Make sure to resize the Collection symbol that was created to show all four (4) tables.

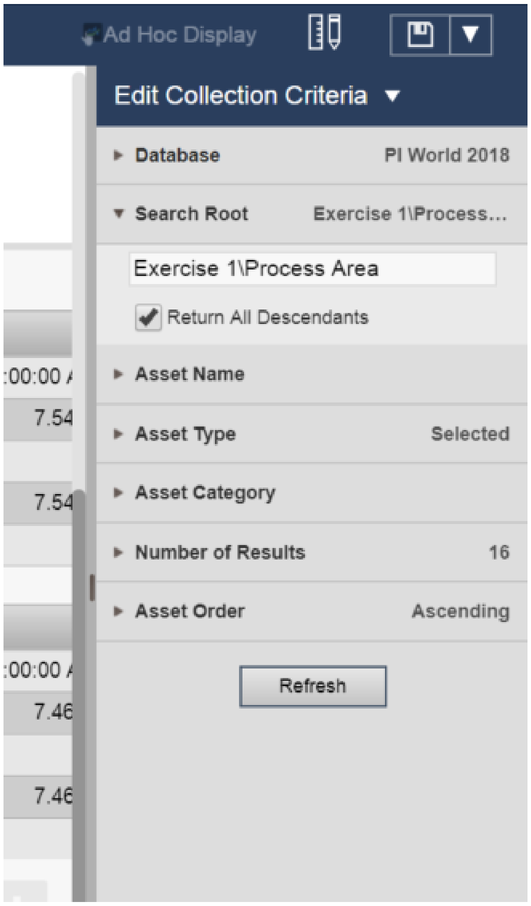
Name	Value	Units
Mixer 1|Last Maintenance Date	3/26/2018 12:00:00 AM	
Mixer 1|MTD Run Hours	55.143	h
Mixer 1|Previous Day Run Hours	16.244	h
Mixer 1|Run Hours Since Maintenanc	14.327	h
Mixer 1|Valve Actuation Count Since	16	count

Name	Value	Units
Mixer 1|Last Maintenance Date	3/23/2018 12:00:00 AM	
Mixer 1|MTD Run Hours	7.5494	h
Mixer 1|Previous Day Run Hours	0	h
Mixer 1|Run Hours Since Maintenanc	7.5494	h
Mixer 1|Valve Actuation Count Since	62	count

Name	Value	Units
Mixer 2|Last Maintenance Date	3/25/2018 12:00:00 AM	
Mixer 2|MTD Run Hours	55.889	h
Mixer 2|Previous Day Run Hours	16.317	h
Mixer 2|Run Hours Since Maintenanc	30.733	h
Mixer 2|Valve Actuation Count Since	34	count

Name	Value	Units
Mixer 2|Last Maintenance Date	3/24/2018 12:00:00 AM	
Mixer 2|MTD Run Hours	7.4606	h
Mixer 2|Previous Day Run Hours	0	h
Mixer 2|Run Hours Since Maintenanc	7.4606	h
Mixer 2|Valve Actuation Count Since	53	count

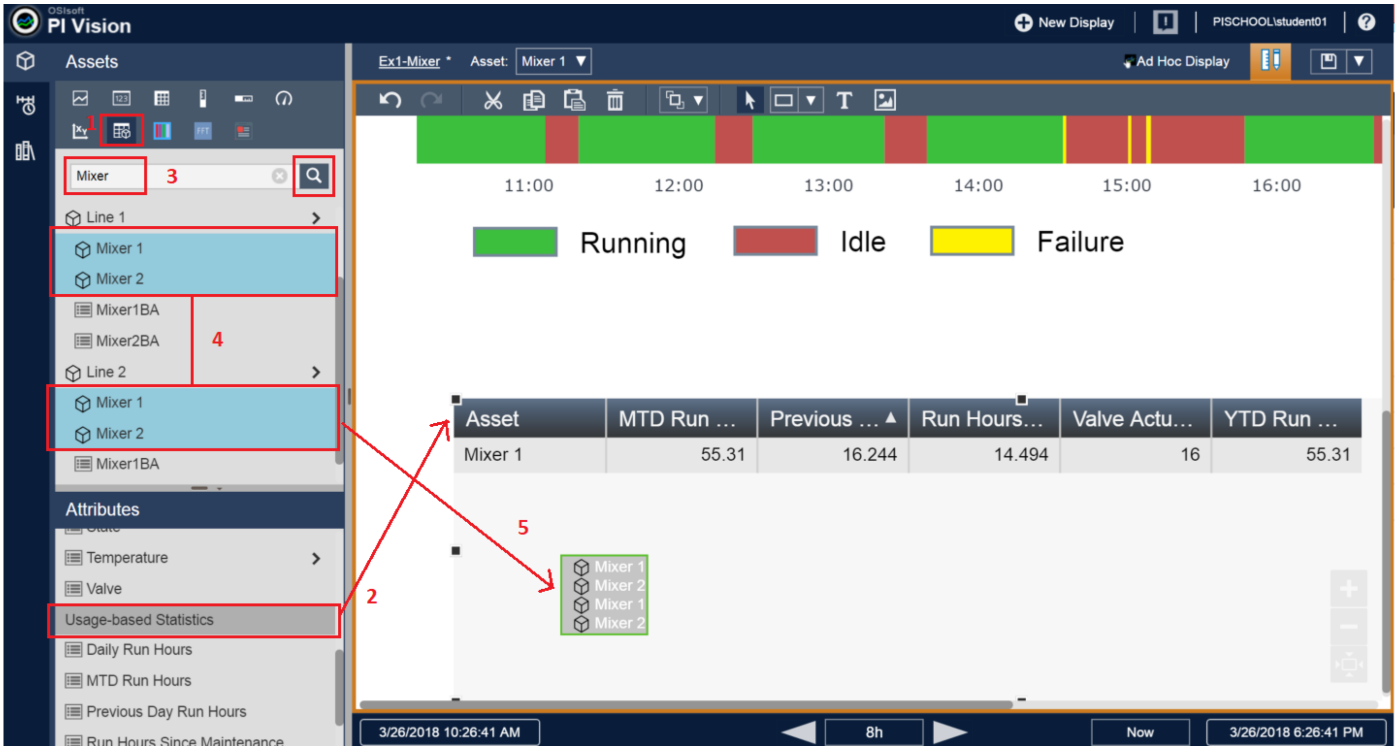
12. In order to show all four mixers, you may need to right-click the Collection that was created, select the **Edit Collection Criteria…** and modify the **Search Root** field like in the below picture (remember to check the **Return All Descendants** option):



Then click the **Refresh** button. Resize the symbol to show all four (4) mixers information.

13. Another way, to compare the statistics, but this time in one table, is to use the **Asset Comparison Table** symbol.

    1. Select the **Asset Comparison Table** symbol ().
    2. Drag the whole **Usage-based Statistics** attribute category on the display.
    3. Search for mixer under the Process Area level in the hierarchy to find all four mixers.
    4. Using the **Ctrl** key, multi-select the four mixers.
    5. Drag the four mixers over the Asset Comparison Table to add all four to the symbol.



Once all four assets are added, it is possible to order by one of the columns by clicking on the column header of interest (make sure the display is not showing the edit symbol ( ) in its upper-right corner to use this functionality).

Asset	Asset ID	MTD Run...▼	Previous D...	Run Hours...	Valve Actu...	YTD Run ...	Daily F
Mixer 2	510	56.055	16.317	30.9	34	56.055	Pt
Mixer 1	509	55.477	16.244	14.661	16	55.477	Pt
Mixer 1	609	7.7722	0	7.7722	62	7.7722	Pt
Mixer 2	610	7.7106	0	7.7106	53	7.7106	Pt

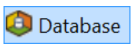
Exercise 2 – Condition-based Maintenance (CbM)

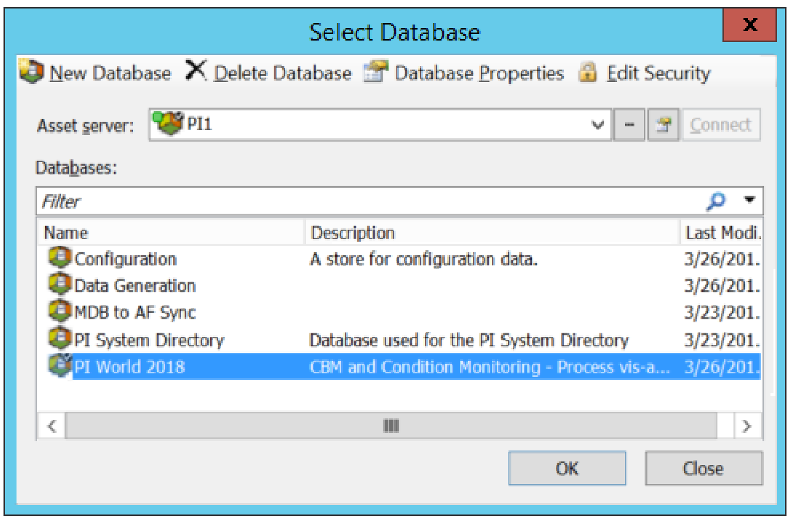
In this exercise, we assess the condition of an equipment by calculating metrics that can serve as leading indicators of equipment failure or loss of efficiency – for example, bearing temperature to understand the pump bearing condition.

We track the alerts for the bearing temperature and then discuss the use of PI Notification to send an email or use the web service delivery channel to notify a system (i.e. triggering a work order) for follow-up action. The bearing temperature events are viewed in a watchlist in PI Vision.

First, look at usage-based information in AF structure used for this exercise.

Step 1: Exploring the Asset Framework structure

1. Open **PI System Explorer** and connect to the **PI World 2018** AF database. If the top bar of the PI System Explorer window does not show \\PI1\PI World 2018 already, then click on the top toolbar  button to select the PI AF database.

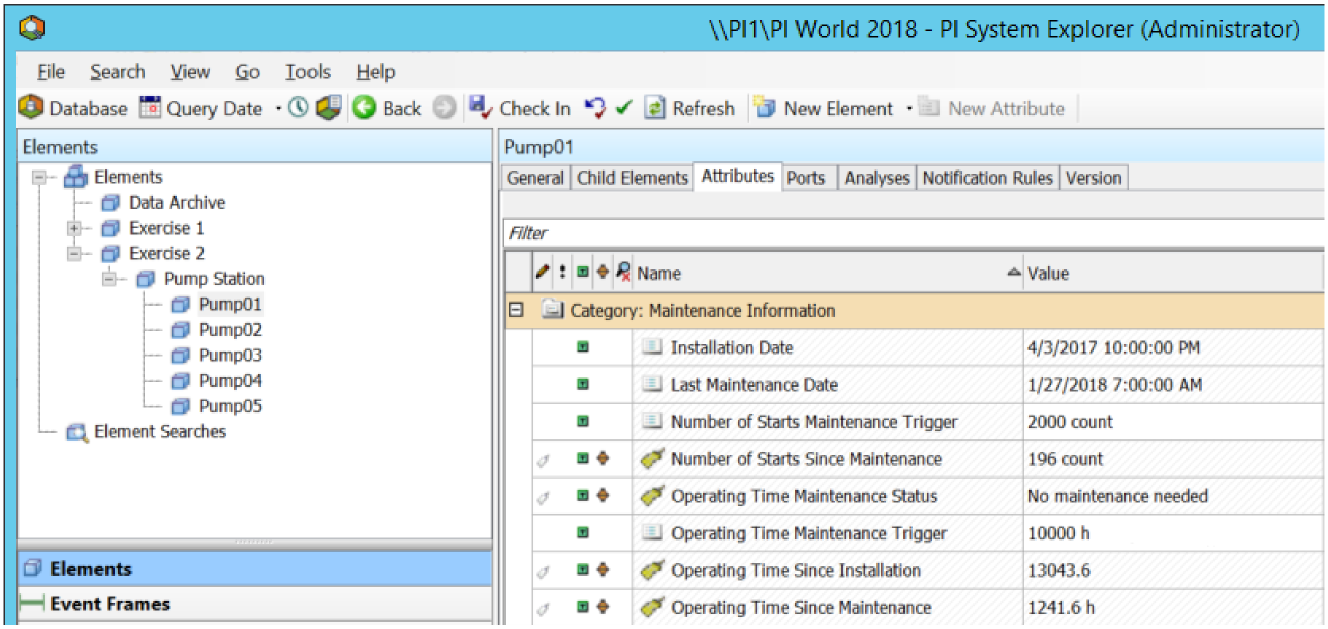


2. From **Elements** section of PI System Explorer, explore the structure for Exercise 2 by drilling-down under **Exercise 2-3 > Pump Station > Pump01**.

This pump station currently has five (5) pumps. A new sixth pump has been ordered and will soon be installed.

Maintenance is important for these critical assets. Usage-based maintenance has already been implemented, in a manner that we saw for the ice-cream mixers in Exercise 1.

3. From the **Attributes** tab of Pump01, explore the **Maintenance Information** section.



Number of starts and operating time (run hours) thresholds have been added and current totals since the last maintenance can be evaluated to determine if maintenance is required based on those criteria.

4. Explore the analysis expression from the **Analyses** tab. The **Usage-based Calculations** analysis evaluates the new totals on a regular basis and compares them to their limits.

Name	Expression

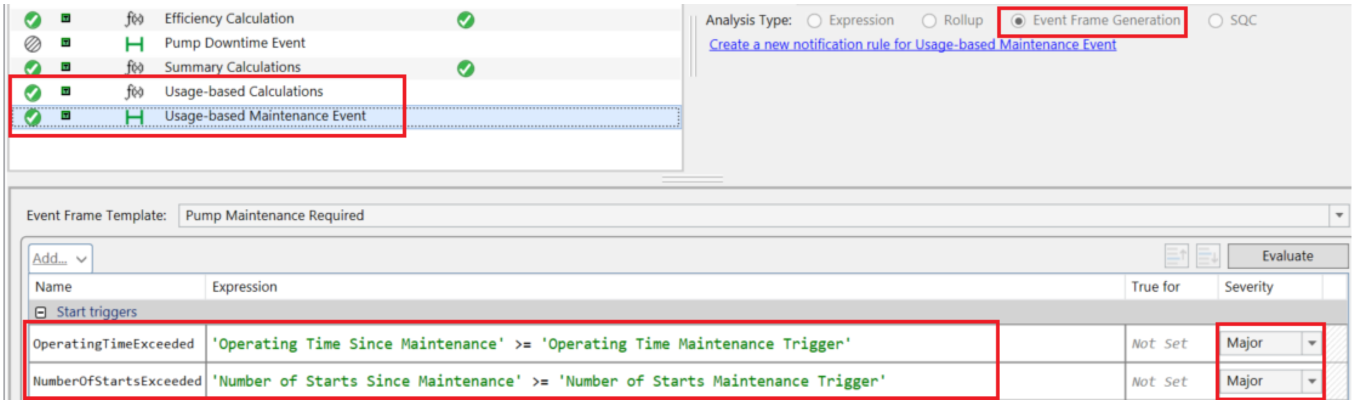
InstallationRuntime	//Hours since the pump was installed
	//TimeEq function results in seconds; need to define that before converting again to hours
	IF 'Installation Date' > '*' THEN 0
	ELSE Convert(Convert(TimeEq('Pump Status','Installation Date','*',"ON"), "s"), "h")

NumberofStarts	//Counts the number of times the value of pump status has changed.
	//Since you have a count for all "on" and "off", divide by 2
	Convert(NumOfChanges('Pump Status','Last Maintenance Date','*')/2, "count")

LastMaintRuntime	//Hours since the last maintenance date
	//TimeEq function results in seconds; need to define that before converting again to hours
	IF 'Last Maintenance Date' > '*' THEN 0
	Else Convert(Convert(TimeEq('Pump Status','Last Maintenance Date','*',"ON"), "s"), "h")

Status	If LastMaintRuntime >= 'Operating Time Maintenance Trigger' Or NumberofStarts >= 'Number of Starts Maintenance Trigger' Then 1 Else 0

In order to keep track of instances where the values violate threshold limits, you can configure analysis to keep track of those as events. Using the **Event Frame Generation** option, events will be generated to track what is relevant to you.
In this example, the usage-based statistics are tracked by the **Usage-based Maintenance Event** analysis.

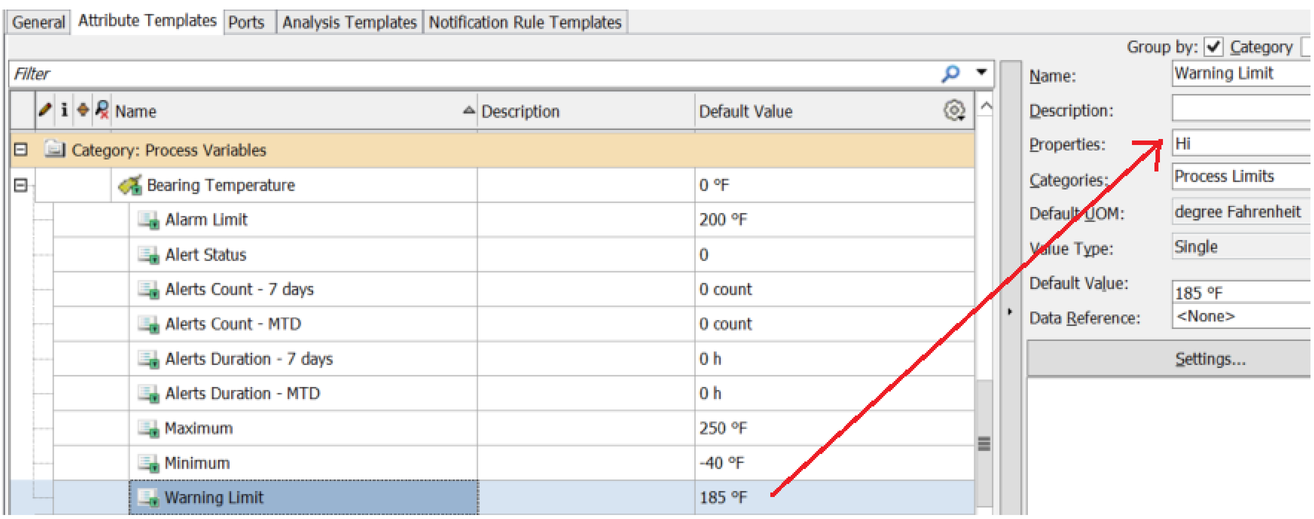


Step 2: Monitoring the bearing temperature

In addition to usage-based metrics, we can also look at sensor data, say, a temperature. Bearing temperatures for the pumps are available.

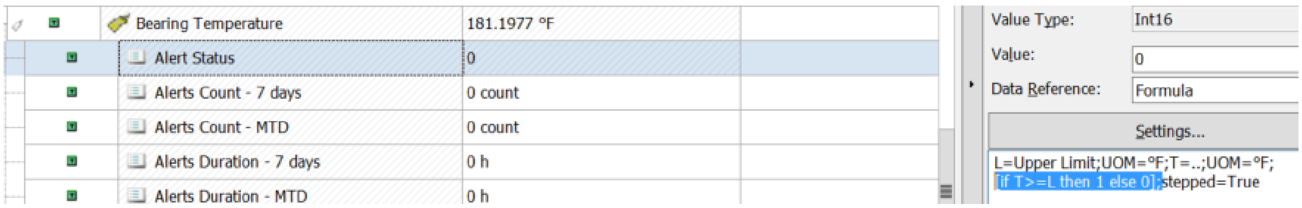
Under normal conditions, we know that the temperature should not exceed a defined high limit.

1. Look under the **Attributes** for Pump01 and expand the attributes under the **Bearing Temperature**.



The **Warning Limit** and **Alarm Limit** attributes define the high limits; they are configured with the attribute trait **Hi** and **HiHi** respectively, as shown in the **Properties** field (this will be used in PI Vision).
Note the presence of placeholders for count and duration of temperature alerts; these will be configured in the steps below.

2. Explore the Formula attribute named **Alert Status**. It is evaluating whether or not the bearing temperature is above its warning limit, and if so turns to 1 (if T>=L then 1 else 0). This will be used to count the number of occurrences and the duration of the alerts in the next step.

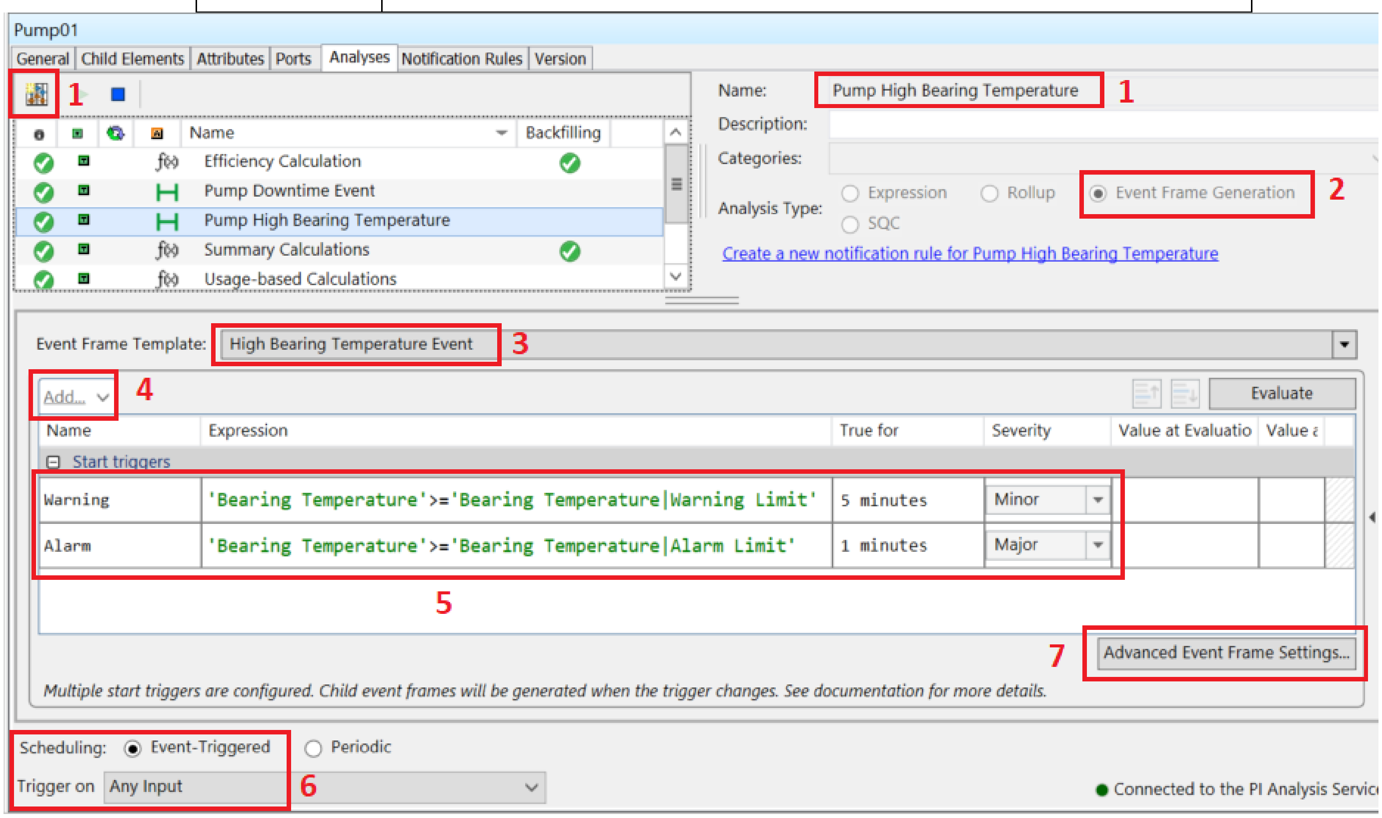


Step 3: Tracking high bearing temperature events

Using Event Frames, we can track the instances where the temperature exceeded its limits.

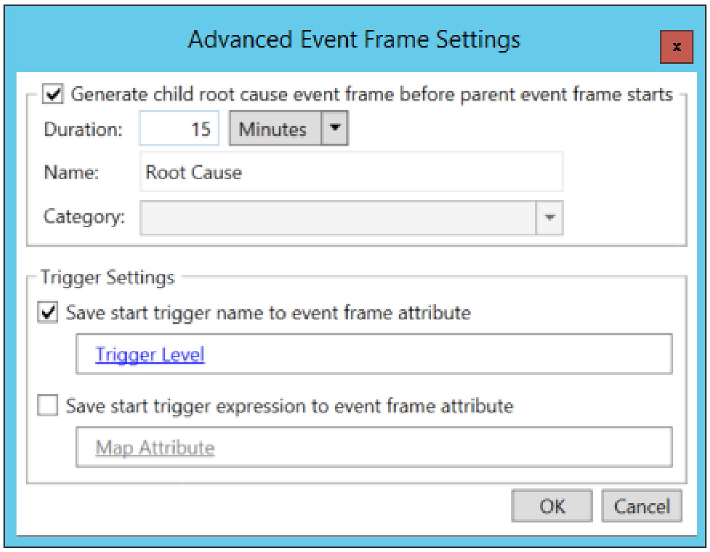
1. Navigate to the **Analyses** tab for Pump01 and create a new analysis named **Pump High Bearing Temperature**.
2. Change the analysis type to **Event Frame Generation**.
3. Select the **High Bearing Temperature Event** event frame template.
4. Add a new **Start Trigger** from the **Add...** link.
5. Configure one Warning-level trigger and one Alarm-level trigger as shown in the below picture.

Name	Expression
Warning	'Bearing Temperature' >= 'Bearing Temperature|Warning Limit'
Alarm	'Bearing Temperature' >= 'Bearing Temperature|Alarm Limit'

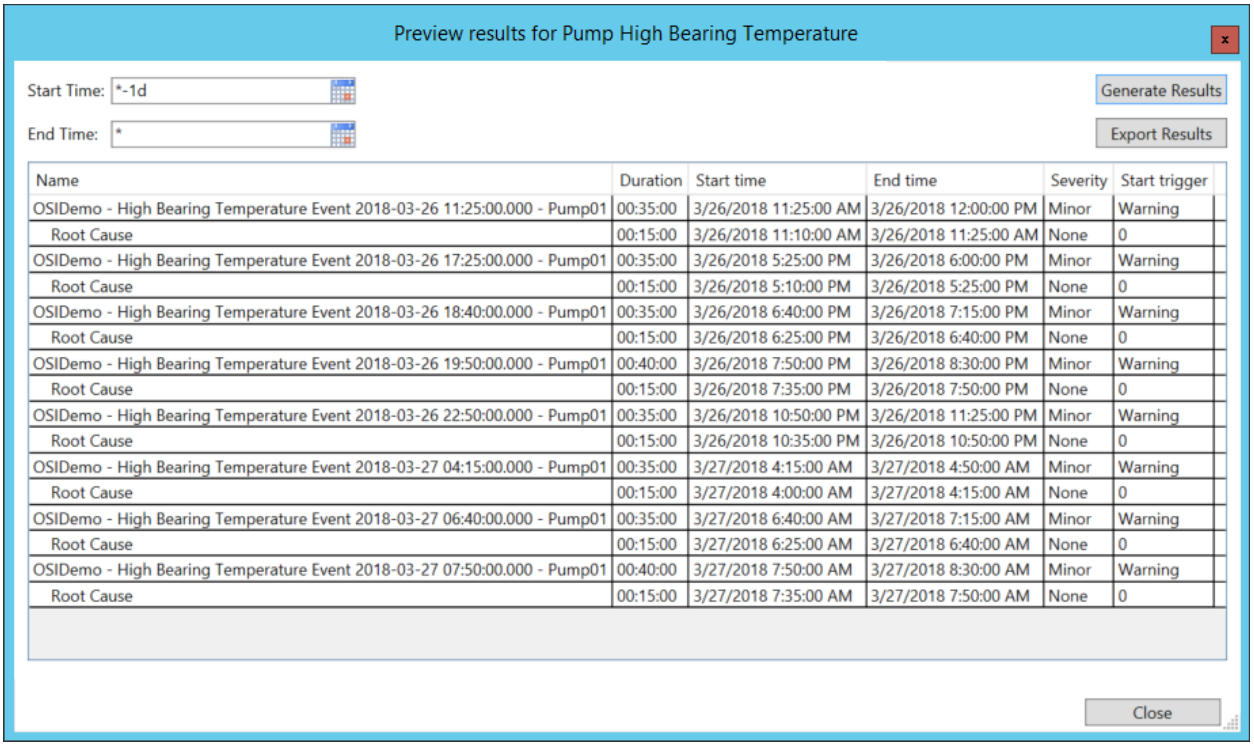


Note: In the current example, the alarm-level trigger needs to be of a higher severity than the warning-level trigger in order to become active. Furthermore, the **True for** (time true) option can be leveraged to make sure valid alerts are being triggered.

6. Leave the scheduling as **Event-Triggered** on **Any Input**.
7. Optionally, you can store the **start trigger name** to an attribute of the event that will get generated from this analysis.
   You can also enable a **Root Cause** child event frame to be generated, in order to have an easy window of time to look at the data (in PI Vision) before the event occurred.



8. Click the Check In button from the top toolbar to save the changes, then right-click the Pump High Bearing Temperature analysis and select **Convert to Template** to enable this analysis for all pumps. Click the Check In button once more.
9. Preview the results for Pump01 by right-clicking the **Pump High Bearing Temperature** analysis and selecting **Preview Results**. You can preview the last day (***-1d** to *****) by pressing the **Generate Results** button to see if any alerts were present.

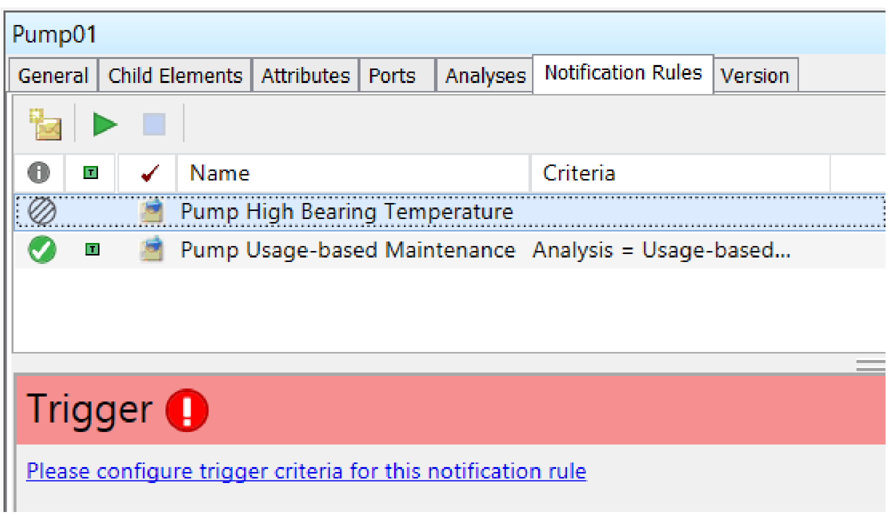


Name	Duration	Start time	End time	Severity	Start trigger
OSIDemo - High Bearing Temperature Event 2018-03-26 11:25:00.000 - Pump01	00:35:00	3/26/2018 11:25:00 AM	3/26/2018 12:00:00 PM	Minor	Warning
Root Cause	00:15:00	3/26/2018 11:10:00 AM	3/26/2018 11:25:00 AM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-26 17:25:00.000 - Pump01	00:35:00	3/26/2018 5:25:00 PM	3/26/2018 6:00:00 PM	Minor	Warning
Root Cause	00:15:00	3/26/2018 5:10:00 PM	3/26/2018 5:25:00 PM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-26 18:40:00.000 - Pump01	00:35:00	3/26/2018 6:40:00 PM	3/26/2018 7:15:00 PM	Minor	Warning
Root Cause	00:15:00	3/26/2018 6:25:00 PM	3/26/2018 6:40:00 PM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-26 19:50:00.000 - Pump01	00:40:00	3/26/2018 7:50:00 PM	3/26/2018 8:30:00 PM	Minor	Warning
Root Cause	00:15:00	3/26/2018 7:35:00 PM	3/26/2018 7:50:00 PM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-26 22:50:00.000 - Pump01	00:35:00	3/26/2018 10:50:00 PM	3/26/2018 11:25:00 PM	Minor	Warning
Root Cause	00:15:00	3/26/2018 10:35:00 PM	3/26/2018 10:50:00 PM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-27 04:15:00.000 - Pump01	00:35:00	3/27/2018 4:15:00 AM	3/27/2018 4:50:00 AM	Minor	Warning
Root Cause	00:15:00	3/27/2018 4:00:00 AM	3/27/2018 4:15:00 AM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-27 06:40:00.000 - Pump01	00:35:00	3/27/2018 6:40:00 AM	3/27/2018 7:15:00 AM	Minor	Warning
Root Cause	00:15:00	3/27/2018 6:25:00 AM	3/27/2018 6:40:00 AM	None	0
OSIDemo - High Bearing Temperature Event 2018-03-27 07:50:00.000 - Pump01	00:40:00	3/27/2018 7:50:00 AM	3/27/2018 8:30:00 AM	Minor	Warning
Root Cause	00:15:00	3/27/2018 7:35:00 AM	3/27/2018 7:50:00 AM	None	0

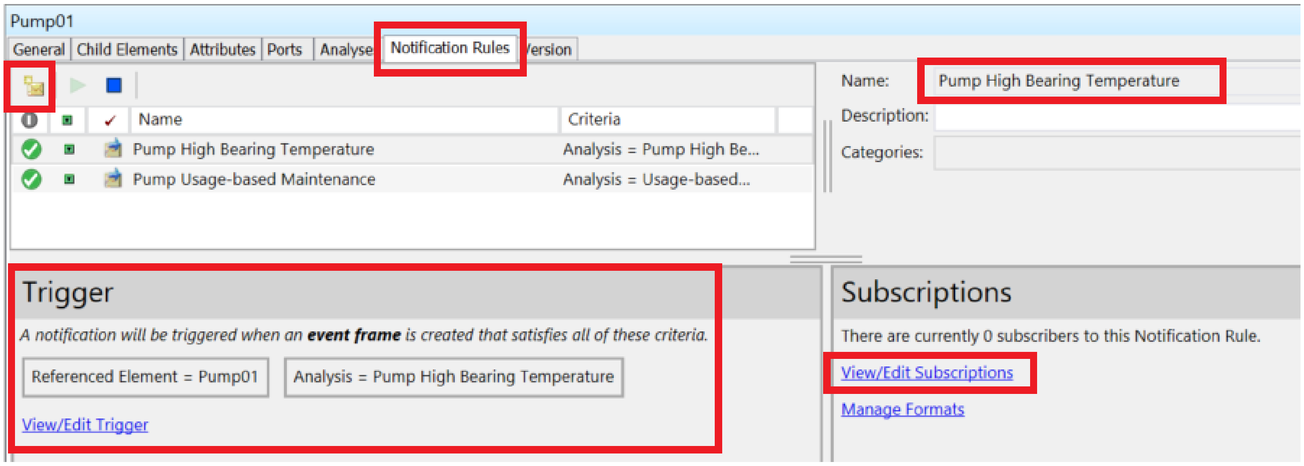
Step 4: Alerting on high bearing temperature events

Events are being tracked by the system, but a Notification can be configured to allow an email to be sent or a web service call to be issued.

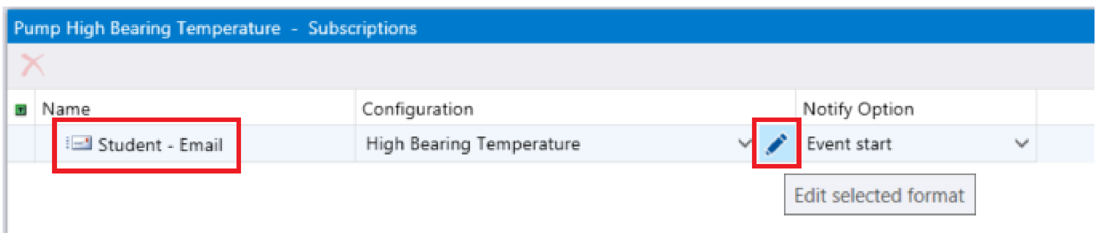
1. Switch to the **Notification Rules** tab of the Pump01 element.
2. Click the **Pump High Bearing Temperature** notification rule.
3. Click the **Please configure trigger criteria for this notification rule** link in the **Trigger** section.



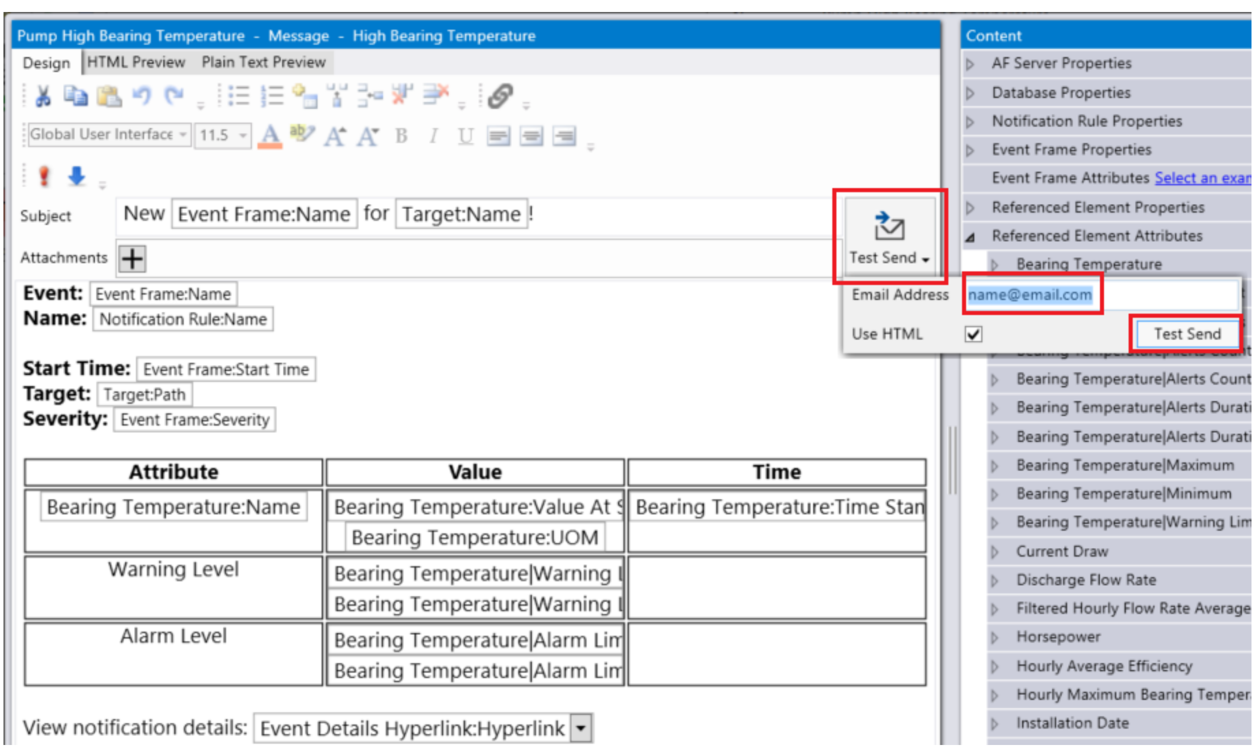
4. Select the **Pump High Bearing Temperature** analysis and leave the rest to default values. Click **OK**.



5. Explore the **Subscriptions** section by clicking the **View/Edit Subscriptions** link on the right.
6. If triggered, the notification will send an email out to the **Student** account, using a format called **High Bearing Temperature**. Click the pencil icon to verify what email format.



7. Before clicking the **Test Send** button, change the email address to an account where you can receive emails – see Step 5 below.



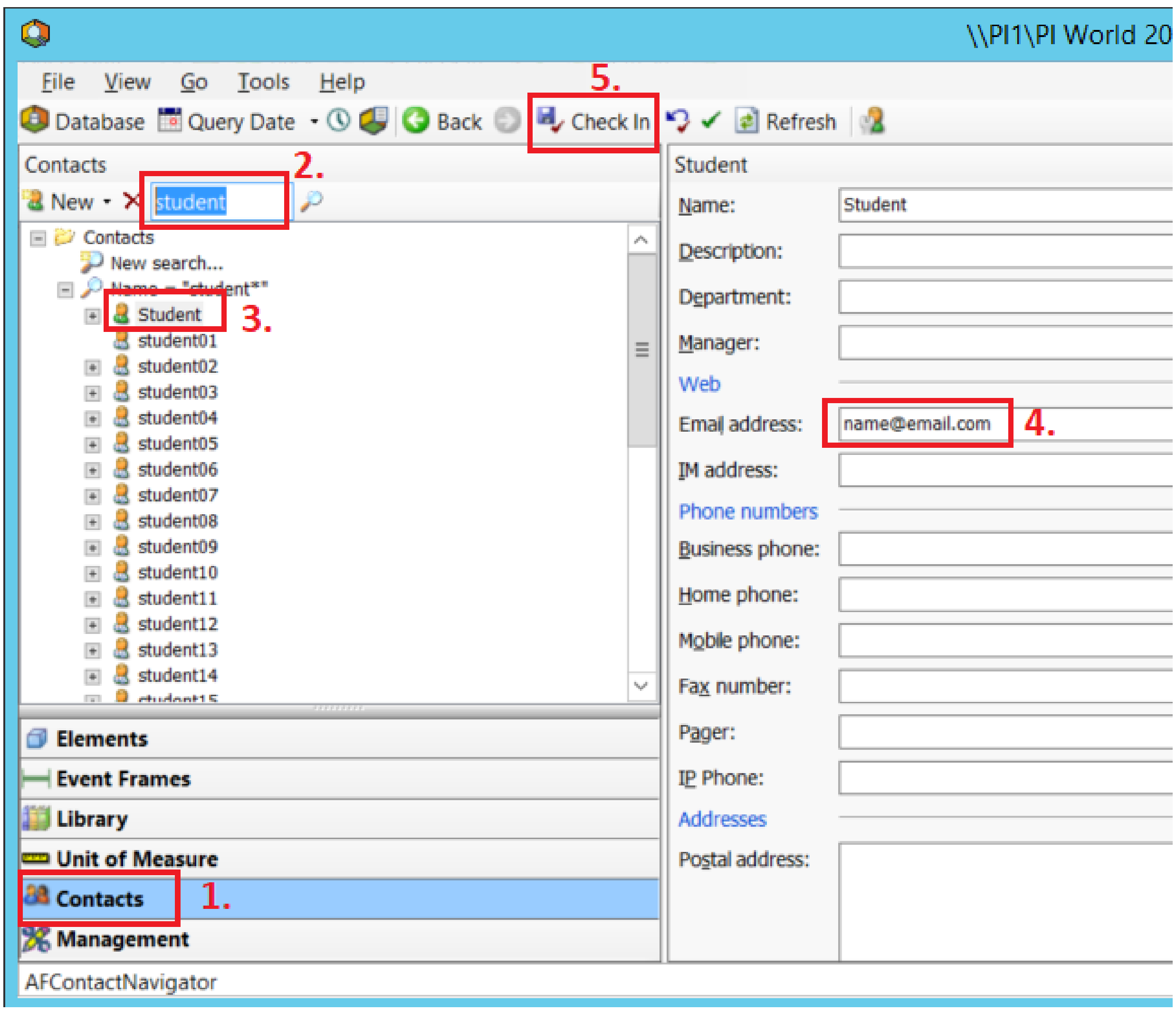
8. Click **Ok** and **Ok** again to exit the windows.
9. Select the **Pump High Bearing Temperature** notification rule and click the **Start** button (▶) to start the rule, and click Check In, then right-click the notification rule to select **Convert to Template** and click Check In again to save the changes.

When a new high bearing temperature event is generated, a notification will be triggered, and an email will be sent out to the Student account's email address.

Step 5: (optional) Sending an email notification on a high bearing temperature alert

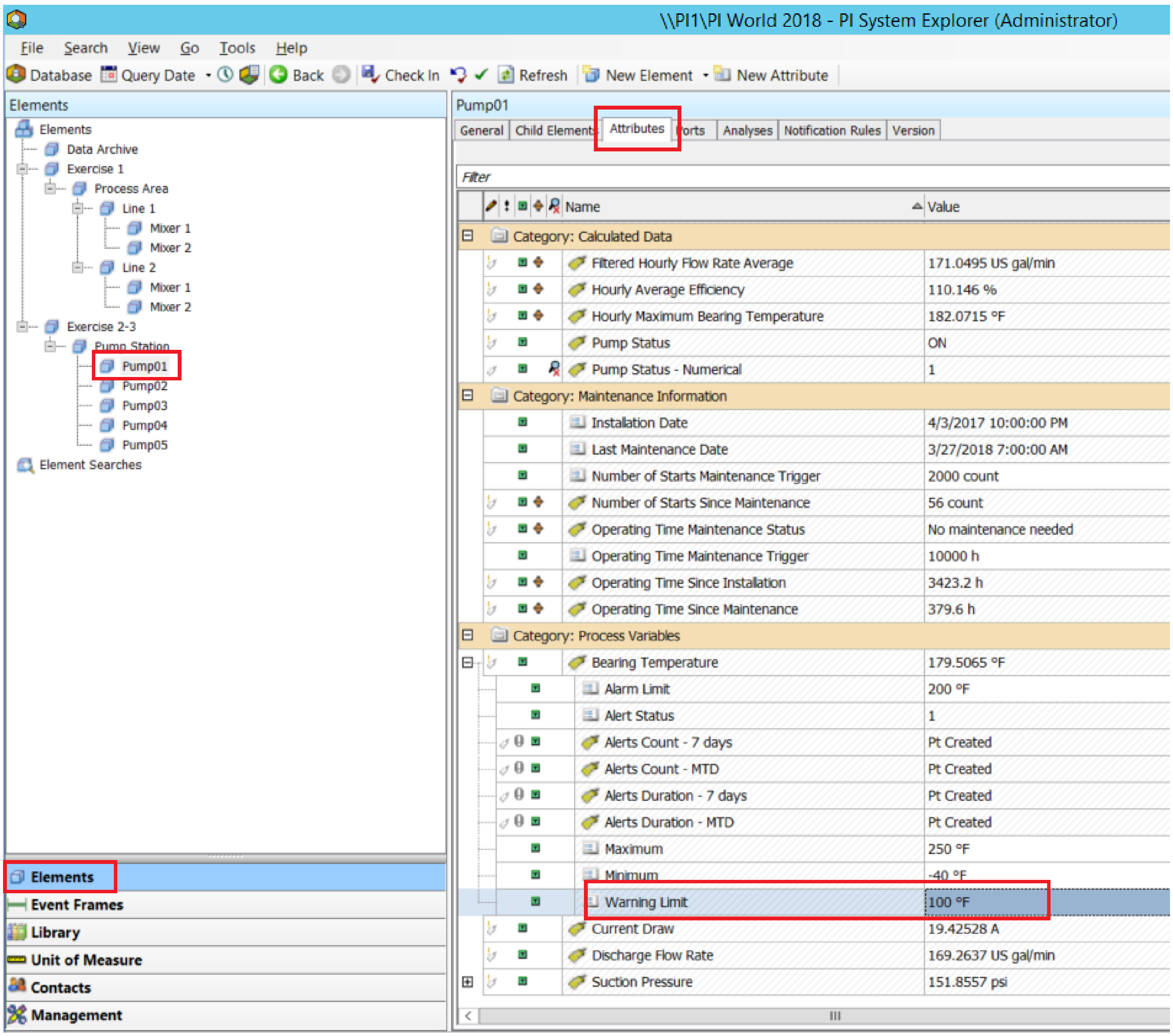
We will first change the email address of the Student to an account that you have access.

1. From PI System Explorer, navigate to the **Contacts** section (refer to the below picture).
2. Search for **student** in the contact search field.
3. Select the **Student** contact.
4. Modify its **Email address** field to reflect an email address you have access to.
5. Click the Check In button to save the changes.



The next triggering of the high bearing temperature should send you an email. Instead of waiting, we will force the temperature to be too high.

6. Navigate to the **Elements** section and to the **Exercise 2-3 > Pump Station > Pump01** element. Select its **Attributes** tab.
7. Expand the **Bearing Temperature** attribute and modify the value of the **Warning Limit** attribute to something lower like **100** °F. Then click the Check In button to save the changes.

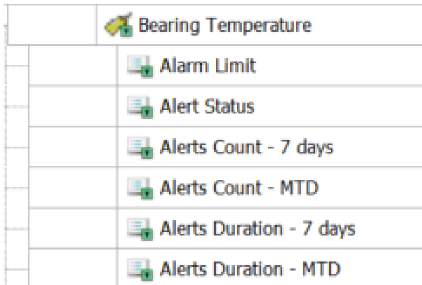


Step 6: Counting number and duration of bearing temperature alerts

For reporting or prioritization purposes, it is useful to have counts on the number of alerts an equipment had over a certain time period.

Total duration of those alerts can also be used to identify equipment most under alert.

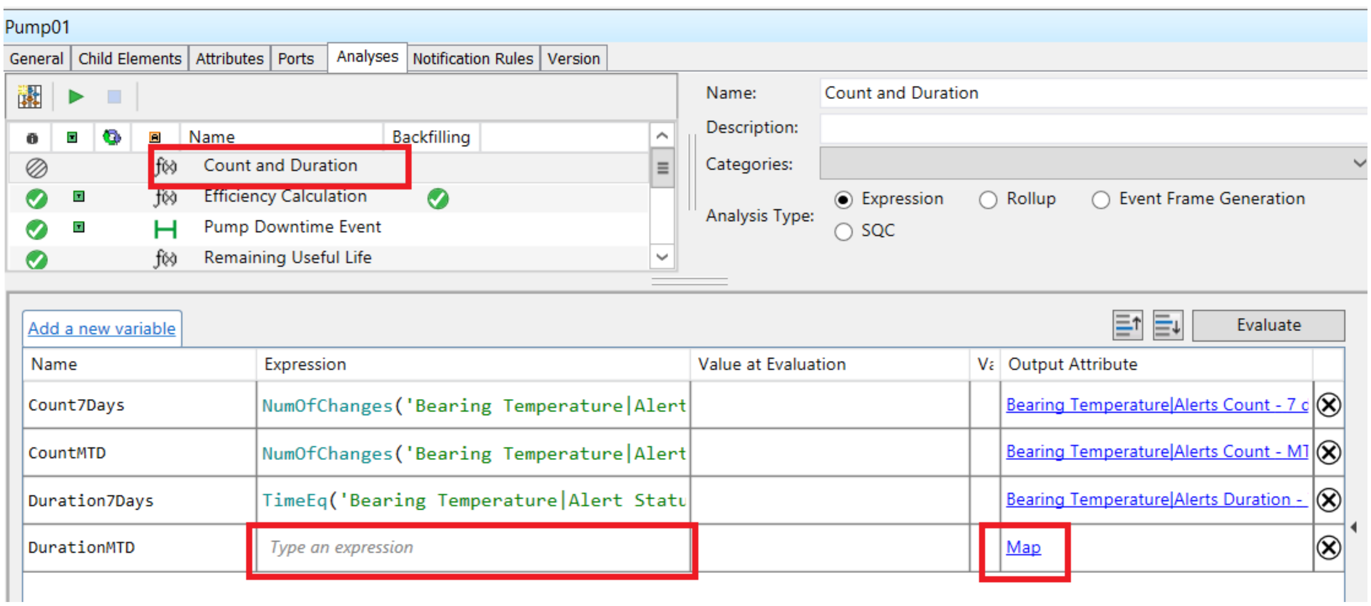
1. We already have placeholders for those counters as part of the **Bearing Temperature** child attributes.



2. Under the **Analyses** tab for **Pump01**, select the **Count and Duration** expression analysis. We only need to complete the fourth variable and start the analysis.
3. Use the **TimeEq()** function to do so on the **Alert Status** attribute to calculate the total duration the bearing temperature was in alert since the beginning of the month.

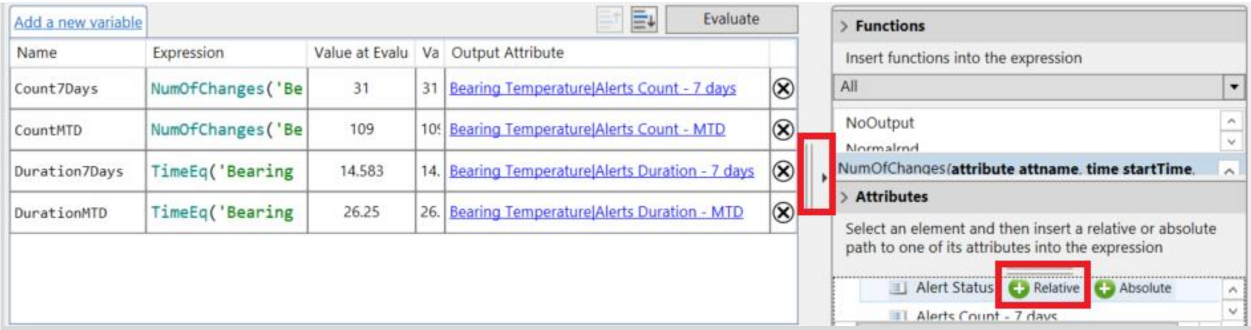
Name	Expression
DurationMTD	`TimeEq('Bearing Temperature|Alert Status','1','*',1)/3600`

4. **Map** the output to the right bearing temperature children attribute.



5. The scheduling is Periodic (5 minutes).
6. Click **Evaluate** to confirm the calculations are functioning properly.

Note: When writing an expression, the auto-complete feature will not suggest child attributes. They can be added from the right-hand side panel:



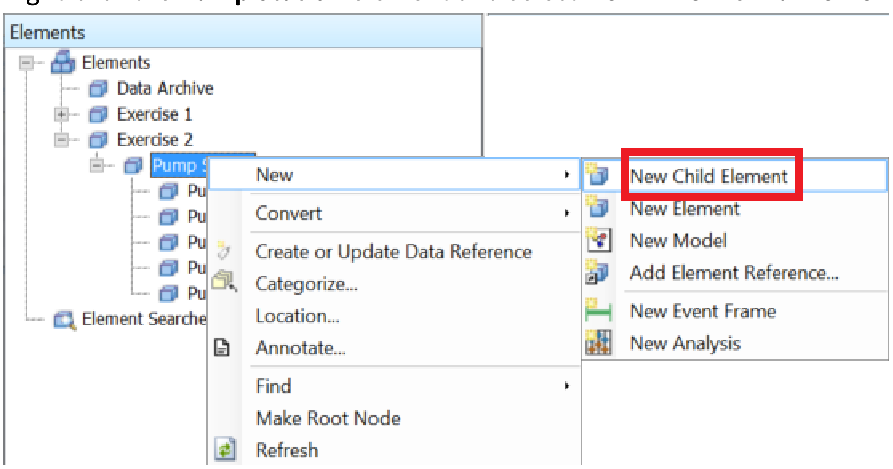
It can also be added by typing the attribute name **'Bearing Temperature|Alert Status'** directly in the expression field.

7. Click Check In, then the **Start** button ( ) to start the analysis, then right-click the analysis to select **Convert to Template** and click Check In again to save the changes.

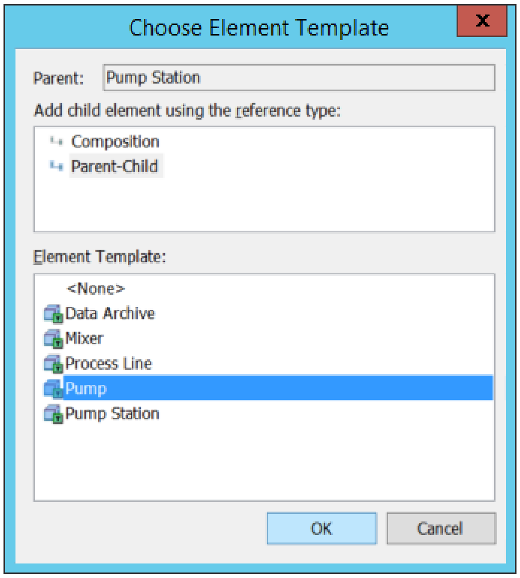
Step 7: (Optional) Adding Pump06

In the current example, we have a Pump template. It includes the attributes, the analyses and notification rules. Pump06 was recently added to the Pump Station and we need to add it to asset structure.

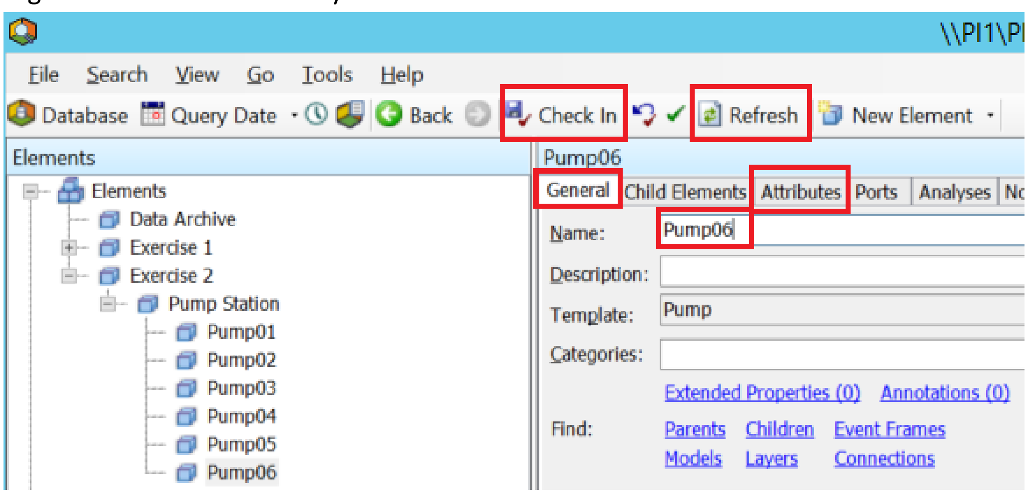
1. Right-click the **Pump Station** element and select **New > New Child Element**.



2. Select the **Pump** template and click **OK**.



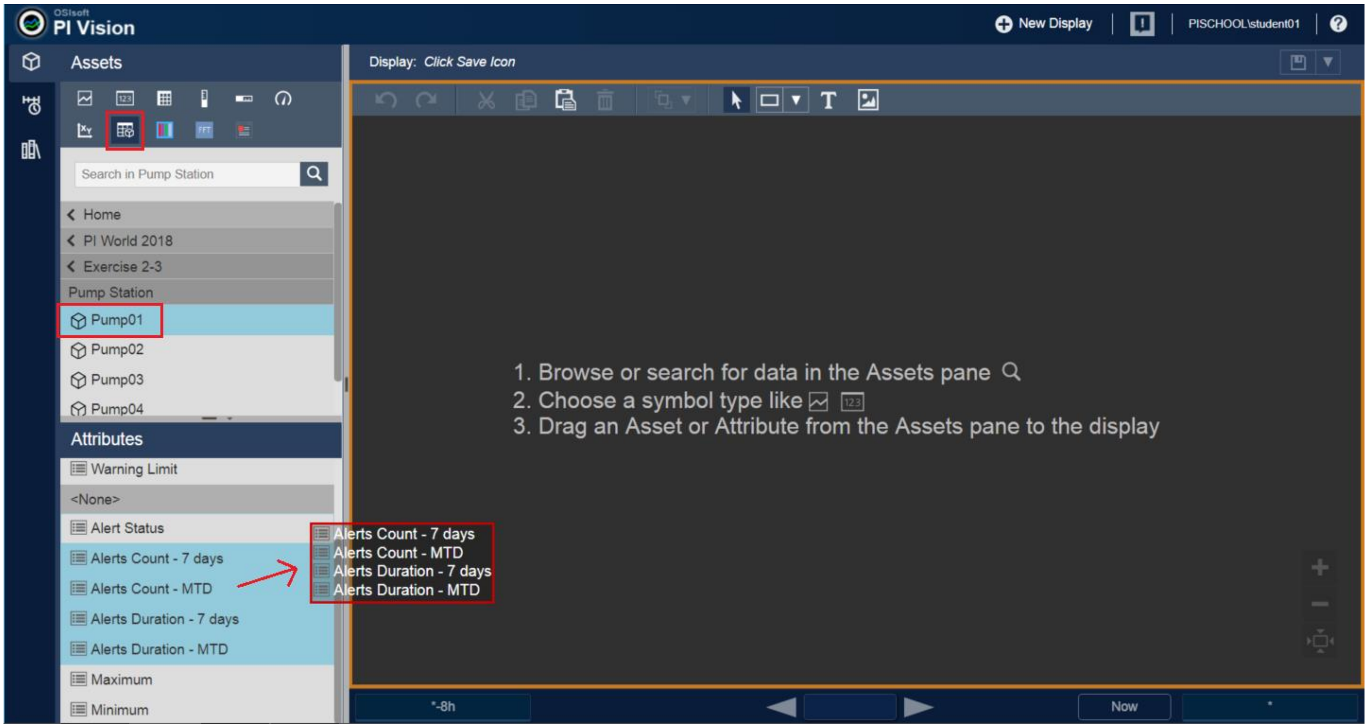
3. Under the **General** tab of the newly created element, rename the element to **Pump06**.
4. Switch to the **Attributes** tab and click the Check In button, and then the **Refresh** button. The tags are found and the analyses started.



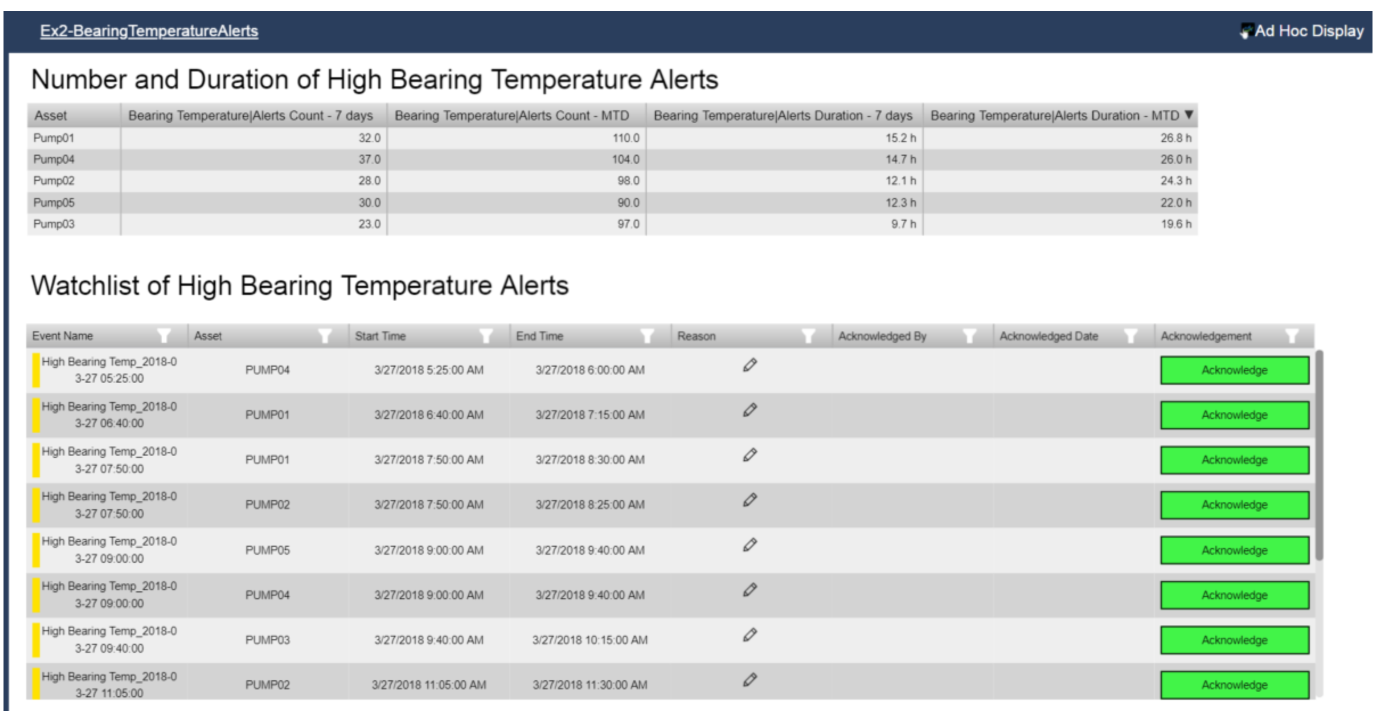
Step 8: (Optional) Visualizing counts and watchlist of events

An easy way to share this information is via PI Vision. Use the below steps to build a new display, or use the existing completed display (https://pi1.pischool.int/PIVision/#/Displays/118/Ex2-BearingTemperatureAlerts).

1. Open the **Google Chrome** web browser from the taskbar and click the **PI Vision** link from the bookmark tool bar or navigate to https://pi1.pischool.int/PIVision.
2. Click the **+ New Display** button in the upper right of the page to create a new display.
3. Use the Assets pane on the left hand side to reach PI World 2018 > Exercise 2-3 > Pump Station > **Pump01**.
4. Select the Asset Comparison Table symbol ( ).
5. Drill-down under the Bearing Temperature attribute from the Attributes section at the bottom.
6. Using the Ctrl or Shift key on your keyboard to multi-select, drag-and-drop the 7-days and MTD count and duration attributes located under the Bearing Temperature attribute.



7. Once you have the proper attributes added for Pump01, drag-and-drop the other pumps elements on the asset comparison table to add extra rows for those other pumps.
8. You can now order the pumps by their MTD hours in alert to understand which asset most under alert condition; in the below example it is Pump01.



Number and Duration of High Bearing Temperature Alerts

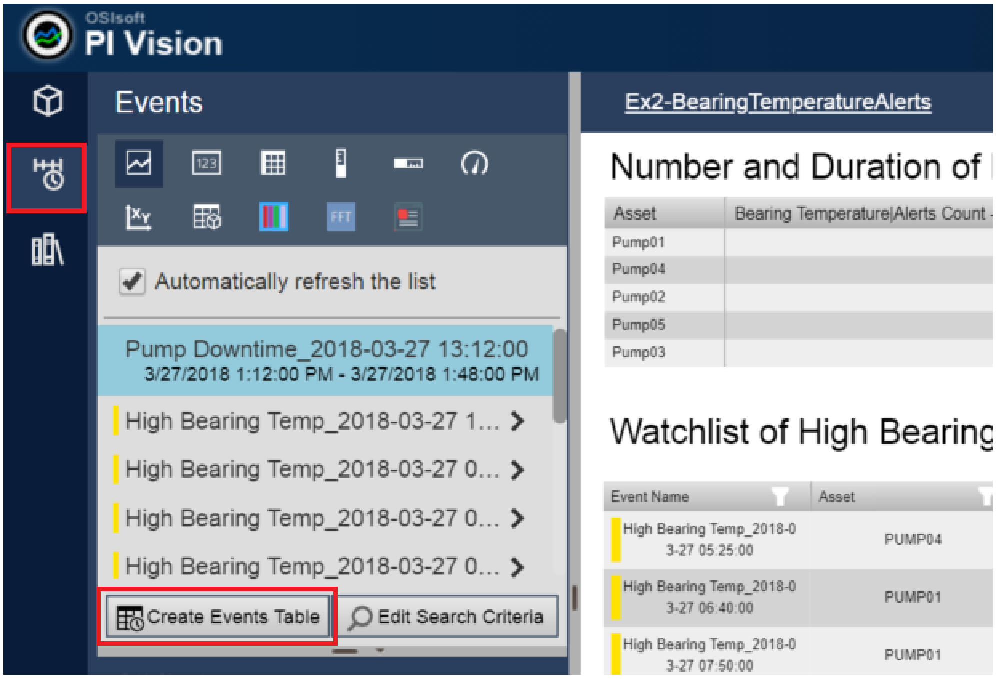
Asset	Bearing Temperature|Alerts Count - 7 days	Bearing Temperature|Alerts Count - MTD	Bearing Temperature|Alerts Duration - 7 days	Bearing Temperature|Alerts Duration - MTD ▼
Pump01	32.0	110.0	15.2 h	26.8 h
Pump04	37.0	104.0	14.7 h	26.0 h
Pump02	28.0	98.0	12.1 h	24.3 h
Pump05	30.0	90.0	12.3 h	22.0 h
Pump03	23.0	97.0	9.7 h	19.6 h

Watchlist of High Bearing Temperature Alerts

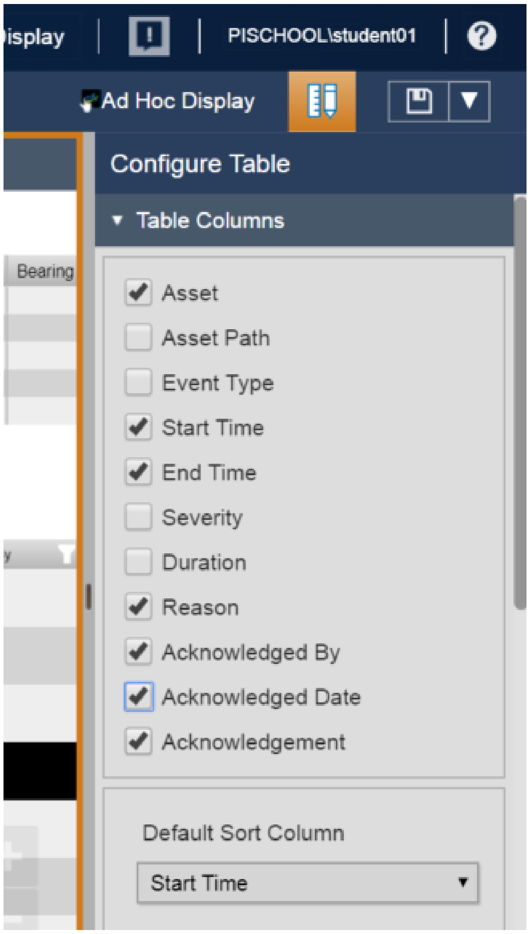
Event Name	Asset	Start Time	End Time	Reason	Acknowledged By	Acknowledged Date	Acknowledgement
High Bearing Temp_2018-03-27 05:25:00	PUMP04	3/27/2018 5:25:00 AM	3/27/2018 6:00:00 AM				Acknowledge
High Bearing Temp_2018-03-27 06:40:00	PUMP01	3/27/2018 6:40:00 AM	3/27/2018 7:15:00 AM				Acknowledge
High Bearing Temp_2018-03-27 07:50:00	PUMP01	3/27/2018 7:50:00 AM	3/27/2018 8:30:00 AM				Acknowledge
High Bearing Temp_2018-03-27 07:50:00	PUMP02	3/27/2018 7:50:00 AM	3/27/2018 8:25:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:00:00	PUMP05	3/27/2018 9:00:00 AM	3/27/2018 9:40:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:00:00	PUMP04	3/27/2018 9:00:00 AM	3/27/2018 9:40:00 AM				Acknowledge
High Bearing Temp_2018-03-27 09:40:00	PUMP03	3/27/2018 9:40:00 AM	3/27/2018 10:15:00 AM				Acknowledge
High Bearing Temp_2018-03-27 11:05:00	PUMP02	3/27/2018 11:05:00 AM	3/27/2018 11:30:00 AM				Acknowledge

Note: You can change the PI Vision display background color by right-clicking the background and selecting **Format Display**.

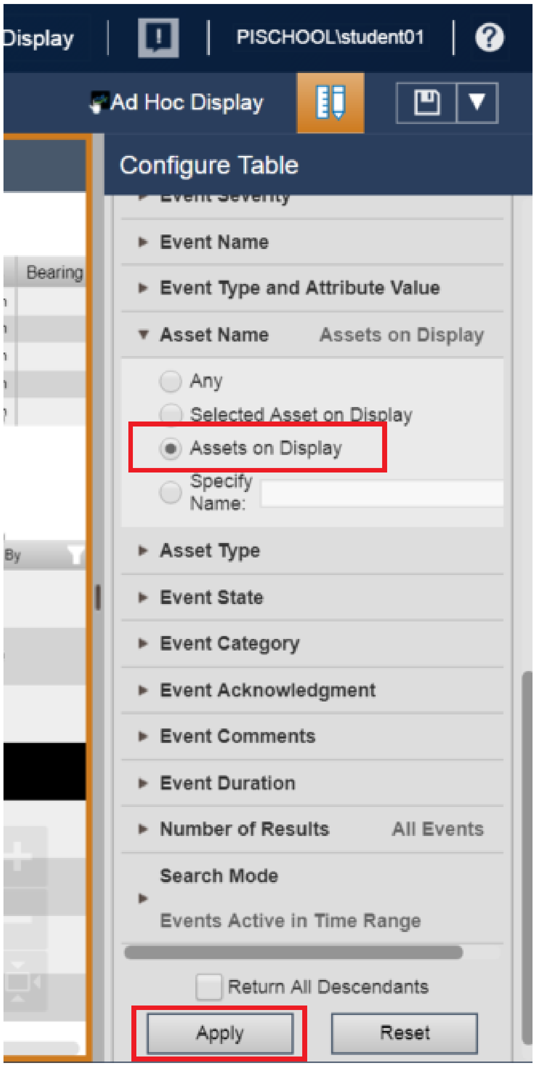
9. Open the left-hand side **Events** pane and click the **Create Events Table** button to add a table that can be used as a watchlist of recent high bearing temperature alerts.



10. Right-click the Events Table to select **Configure Table**.
11. From the configuration pane that shows up, check to add the **Reason**, **Acknowledged By**, and **Acknowledged Date** columns.



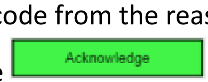
12. Make sure the **Asset Name** field has the **Assets on Display** radio button selected.
13. Click the **Apply** button at the bottom to confirm the changes.



14. Resize the table so it shows all columns correctly.

Step 9: (Optional) Acknowledging and entering reason code for the alerts

The bottom events table shows the active high bearing temperature events in the time range, defined by the display time-bar at the bottom of the screen. The configuration of those events allows you to acknowledge it, as well as to enter a reason code. Both functionalities can be performed from the event table symbol. Acknowledging is useful when sending Notifications, where you need to confirm that they are aware of the alert. The reason code value can leverage a reason tree to specify the cause of the alert, if available.

1. Select one recent event and enter a reason code by clicking the  button and selecting a reason code from the reason tree.
2. Click the 

 button and confirm you can see your username and acknowledgement time after you clicked it.

Event Name	Asset	Start Time	End Time	Reason	Acknowledged By	Acknowledged Date	Acknowledgement
High Bearing Temp_2018-03-27 06:40:00	PUMP01	3/27/2018 6:40:00 AM	3/27/2018 7:15:00 AM	Reason4	PISCHOOL\student01	3/27/2018 2:18:11 PM	Acknowledged

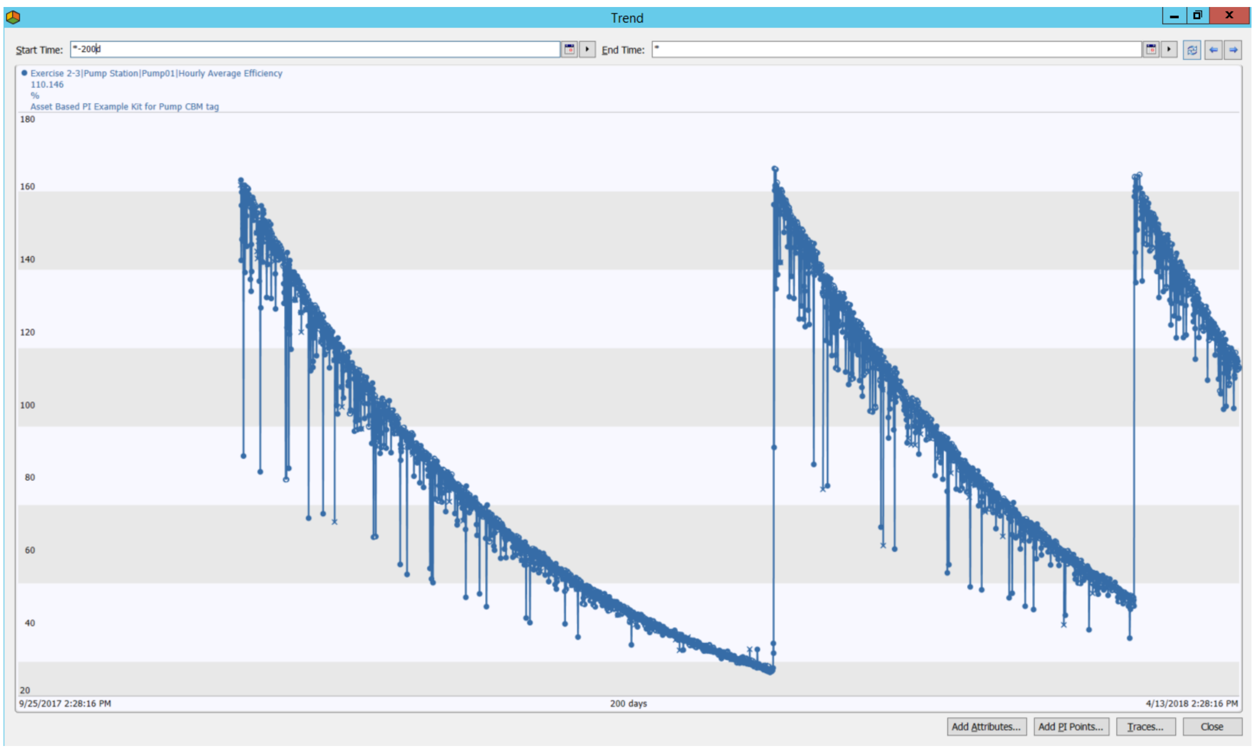
Exercise 3 – Predictive Maintenance (PdM)

Exercise 3a – Simple Predictive - Compressor efficiency

For certain classes of process equipment, their condition can be evaluated by monitoring some key metrics, such as efficiency for a compressor, fouling for a heat-exchanger etc. Often, these metrics show a pattern with time – and, linear, piece-wise linear or non-linear trend which can be extrapolated to estimate remaining-useful-life.

Step 1: Explore the Pump01 efficiency attribute

1. Navigate to the **Elements** section - to **Exercise 2-3 > Pump Station > Pump01**.
2. In the **Attributes** tab, the **Hourly Average Efficiency** attribute is calculated from an analysis named **Efficiency Calculation** (listed in Analyses tab).
3. From the **Attributes** tab, right-click the **Hourly Average Efficiency** attribute and select the **Trend** option. Modify the time range to visualize multiple days, for example from ***-200d** to *****.



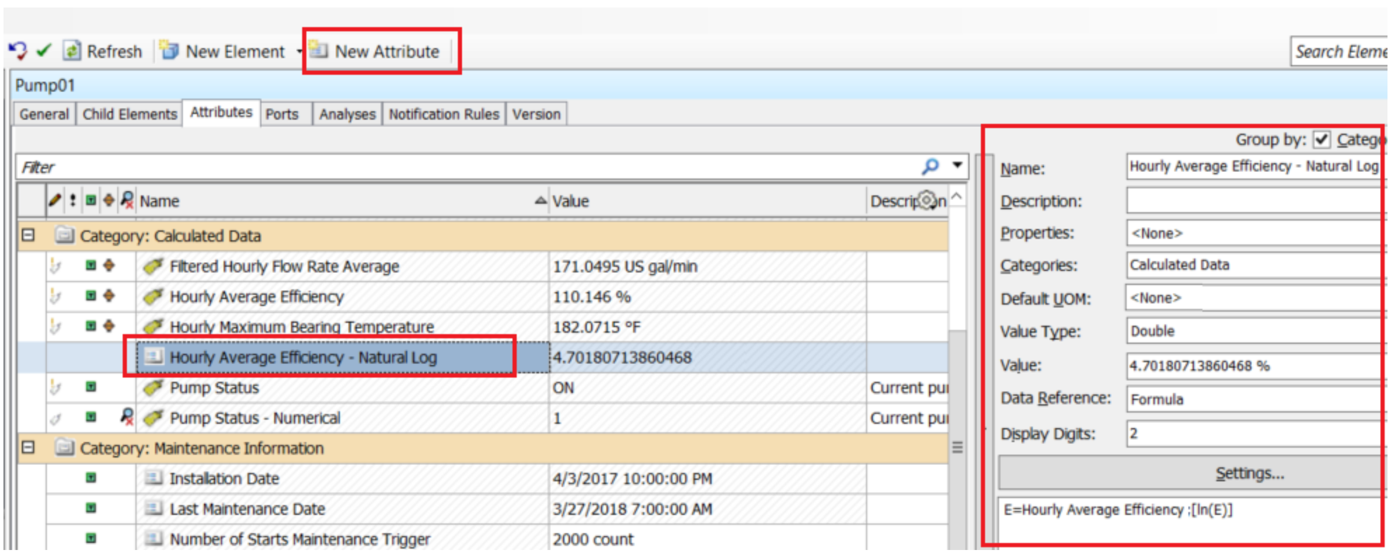
This is not real-life pump data, but rather generated data for this lab. However, efficiency or performance curves can look similar to the above.

Step 2: Create a natural log attribute of the efficiency

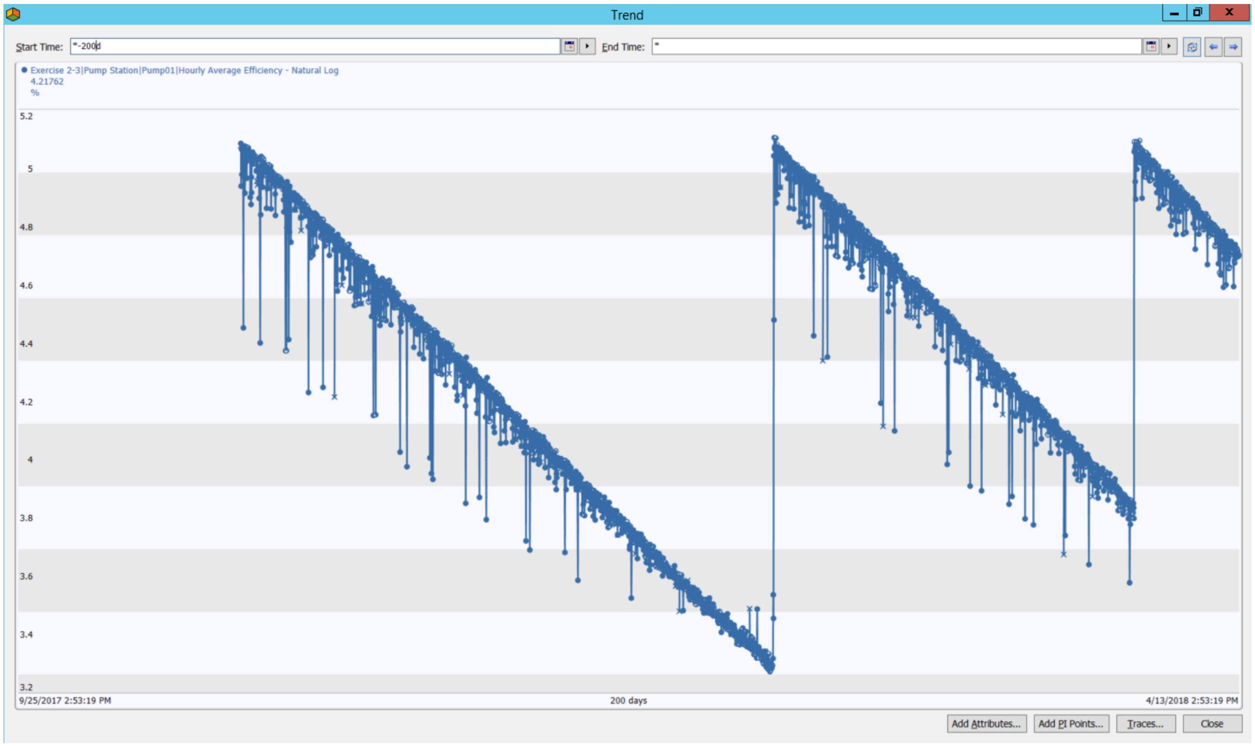
In order to have an easier way to estimate the remaining useful life using the efficiency attribute, we can transform the curve into a linear slope using the natural log or ln().

1. From the **Attributes** tab, click the **New Attribute** button from the top toolbar
2. Set the attribute name = **Hourly Average Efficiency - Natural Log**
3. Set the attribute category = **Calculated Data**
4. Type=**Double**, and use a **Formula** data reference with the following equation:

   E=Hourly Average Efficiency;[ln(E)]



5. Right-click the newly created attribute and select the **Trend** option to validate that the curve resembles a straight slope.

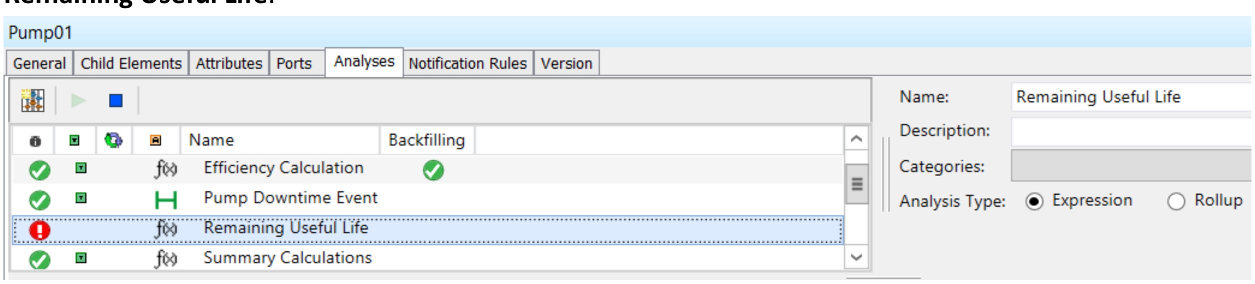


6. Right-click the attribute again to select the **Add Attribute to Template...** option to add the attribute to all pumps.
7. Click the Check In button from the top toolbar to save the changes.

Step 3: Calculate the Remaining Useful Life (RUL) for the Pump01

Use the linear regression function to calculate the slope and thus estimate when the efficiency curve would cross a defined limit to give insight as to its remaining useful life, i.e. schedule maintenance.

1. Locate the **Remaining Useful Life** attribute from the **Attributes** tab of **Pump01**, under the Calculated Data category.
2. Navigate to the **Analyses** tab of Pump01 and create a new Expression analysis named **Remaining Useful Life**.



3. We will use the linear regression **LinRegr()** function to calculate the slope and intercept of the **Hourly Average Efficiency - Natural Log** attribute curve (use the below picture to assist you). A time range of the last 7 days is used to calculate the slope, unless the last maintenance date is within the last 7 days but could also be of various lengths depending on the nature of the data used.

```
LinRegr('Hourly Average Efficiency - Natural Log','*-7d','*')
```

Note:

The LinRegr() function returns an array of three (3) values, the slope, the intercept, and the R^2 value.

In order to extract the information of only one parameter of the array, we can use the result variable and the [x], where x is the index of the parameter you wish to extract.

Refer to the function definition from the right-side panel for more details.

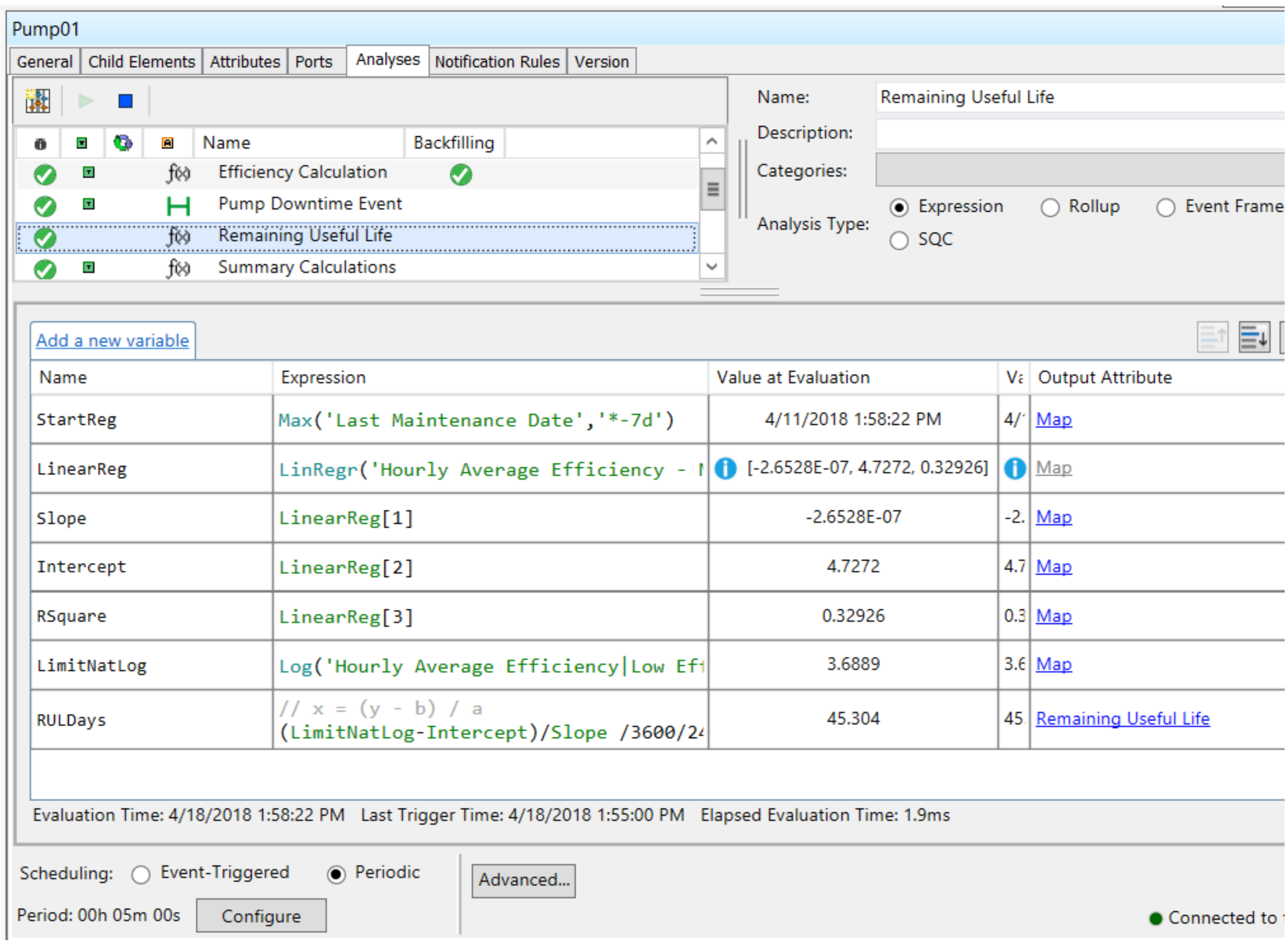
Once we have the slope (often referred to as "a" or "m"), and the intercept (often referred to as "b"), we can consider representing the trace with the *y = ax + b* expression and can determine the time (x) where the trace would cross a certain limit of low efficiency (y).
We will need to transform the Low Efficiency limit using the log() function (natural log), like we initially did for the Hourly Average Efficiency attribute.

4. Use the parameters taken from the result of the LinRegr() expression and the Low Efficiency attribute to calculate the time (x) when this limit will be attained. This will be given by:

$$x = (y - b) / a$$

5. Map this result to the **Remaining Useful Life** attribute.

6. Schedule this calculation to trigger periodically, every **5 minutes**.



Usually, RSquare values less than 0.8 indicate a poor fit and may be due to the curvature assumption and/or noise in the data

Name	Expression
StartReg	`Max('Last Maintenance Date','*-7d')`
LinearReg	`LinRegr('Hourly Average Efficiency - Natural Log',StartReg,'*')`
Slope	`LinearReg[1]`
Intercept	`LinearReg[2]`
RSquare	`LinearReg[3]`
LimitNatLog	`Log('Hourly Average Efficiency|Low Efficiency')`
RULDays	`// x = (y - b) / a` `(LimitNatLog-Intercept)/Slope /3600/24`

7. Click the Check In button from the top toolbar to save the changes. Right-click the Remaining Useful Life analysis to select **Convert to Template**, and then click the Check In button again to save the changes.

8. Navigate back to the **Attributes** tab and validate that you now have a value for the **Remaining Useful Life** attribute of Pump01 (may take up to 5 minutes before a value is written).

Step 4: Compare remaining useful life for all pumps

1. Navigate to the PI Vision display named **Ex3-Remaining Useful Life** by opening the Google Chrome web browser, clicking the PI Vision shortcut from the top favorite toolbar, and then clicking on the right display. Alternatively, navigate to the display using this link: https://pi1.pischool.int/PIVision/#/Displays/124/Ex3-Remaining-Useful-Life

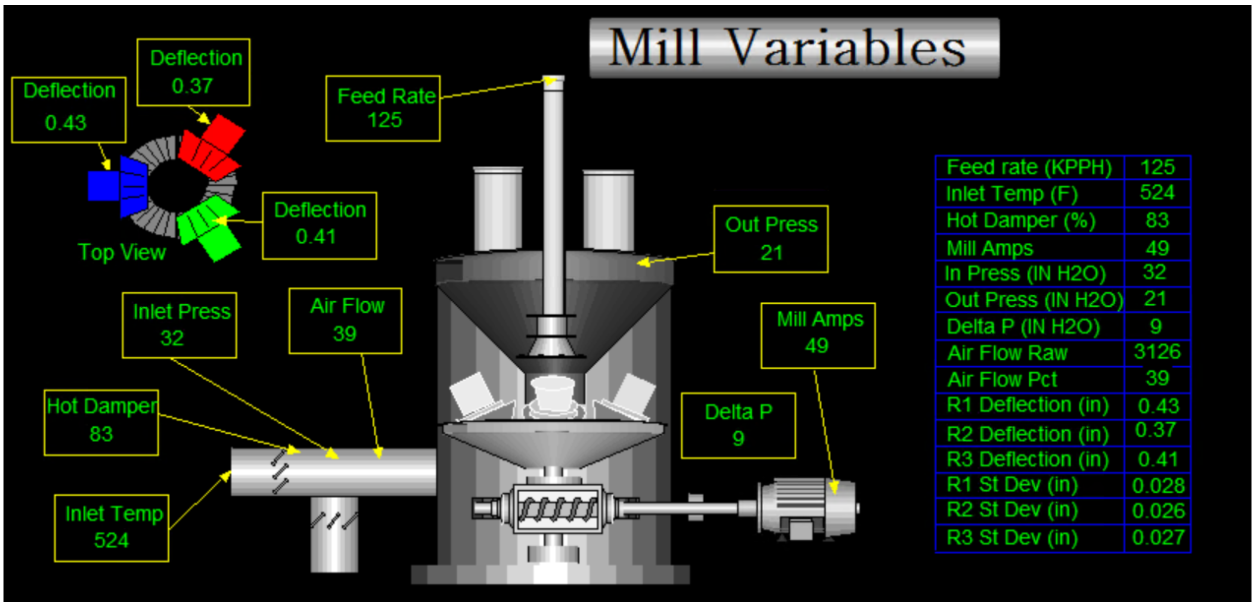
2. Note the remaining useful life in **days** is displayed in the Asset Comparison symbol. You can click on the column header for the remaining useful life to sort the column.

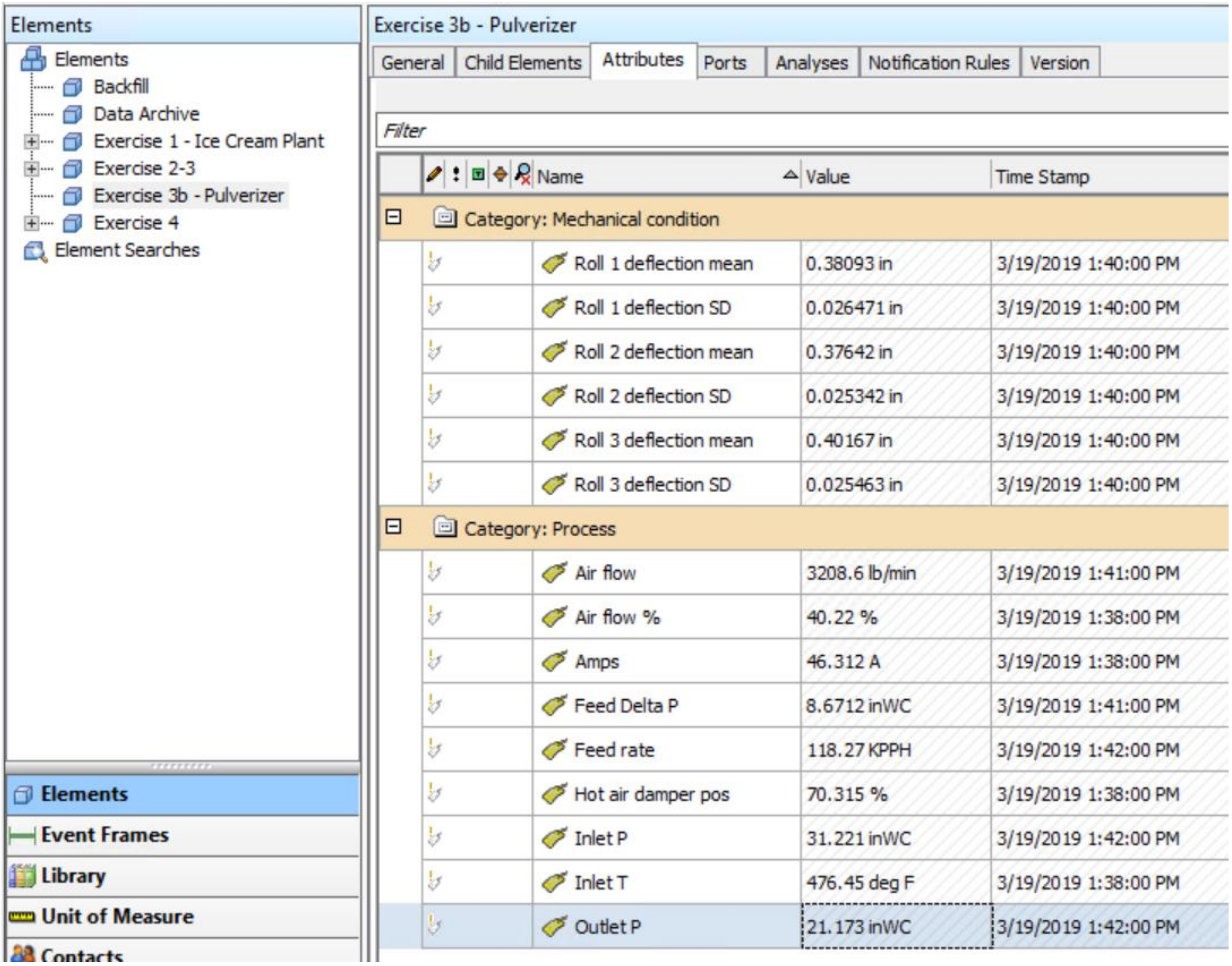
Exercise 3b – Coal Pulverizer – APR (Advanced Pattern Recognition)

In this exercise, we use OSIsoft Partner [ECG's] APR (Advanced Pattern Recognition) based **Predict-It** to monitor a coal pulverizer in a power generation plant.

APR uses similarity based modeling to compare the Pulverizer's current operations with its historical data and detects subtle changes in its run-time behavior to provide early warning fault detection.

Step 1: Pulverizer overview

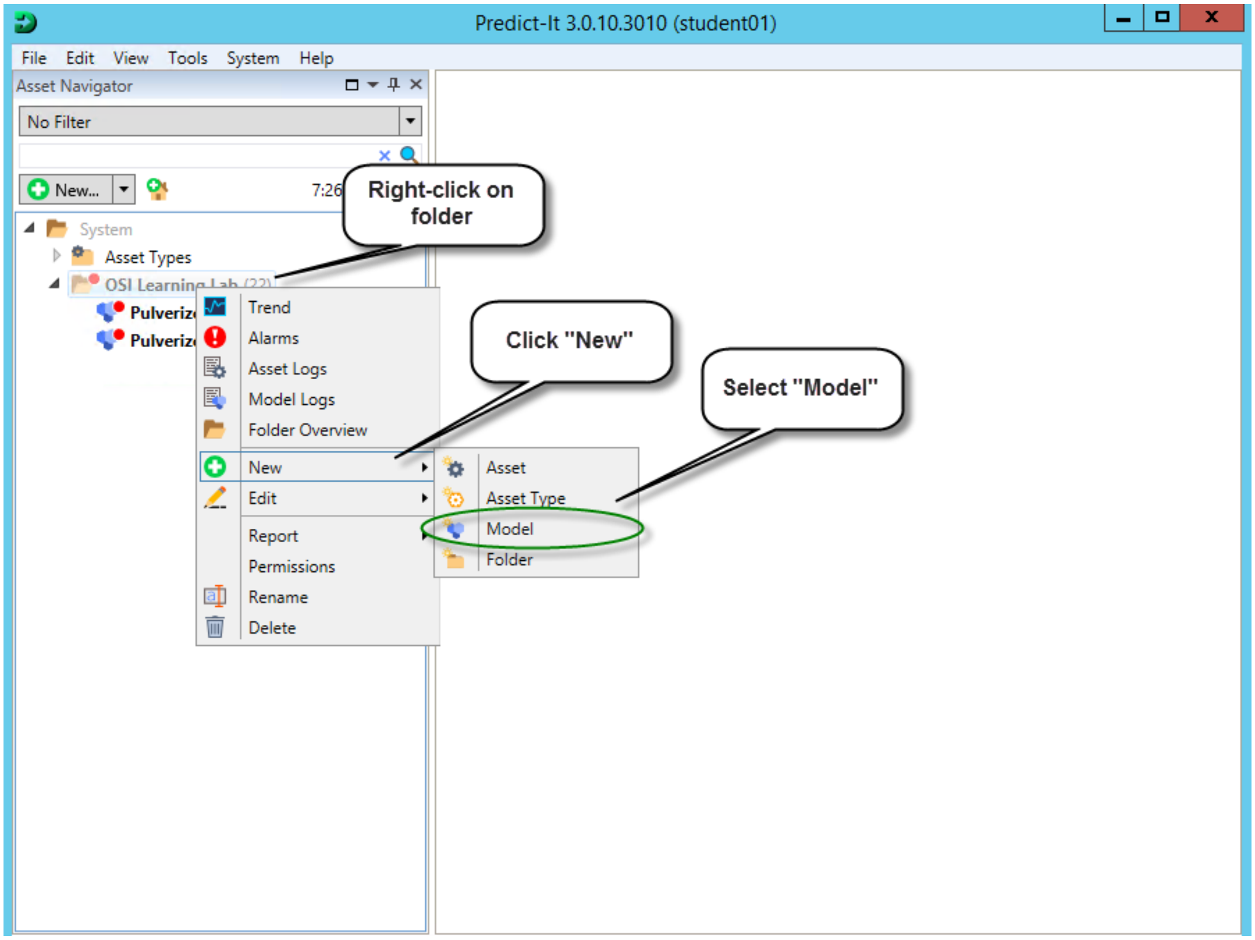




Step 2: Create a new model

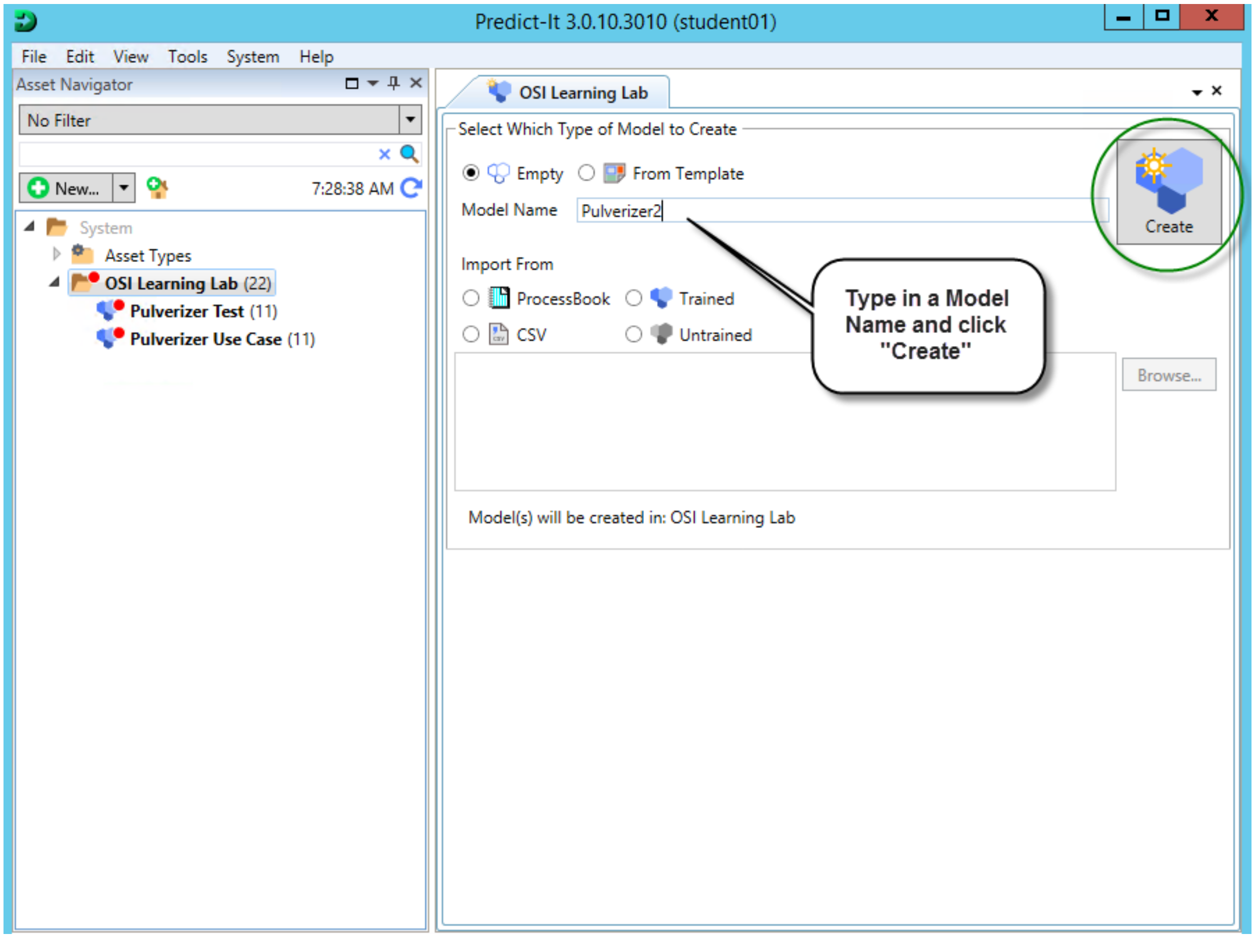
1. Open **Predict-It** overview screen. 

2. Right click on the **PI World 2019** folder to open the options menu. Select **New** and **Model** to bring up the New Model screen.



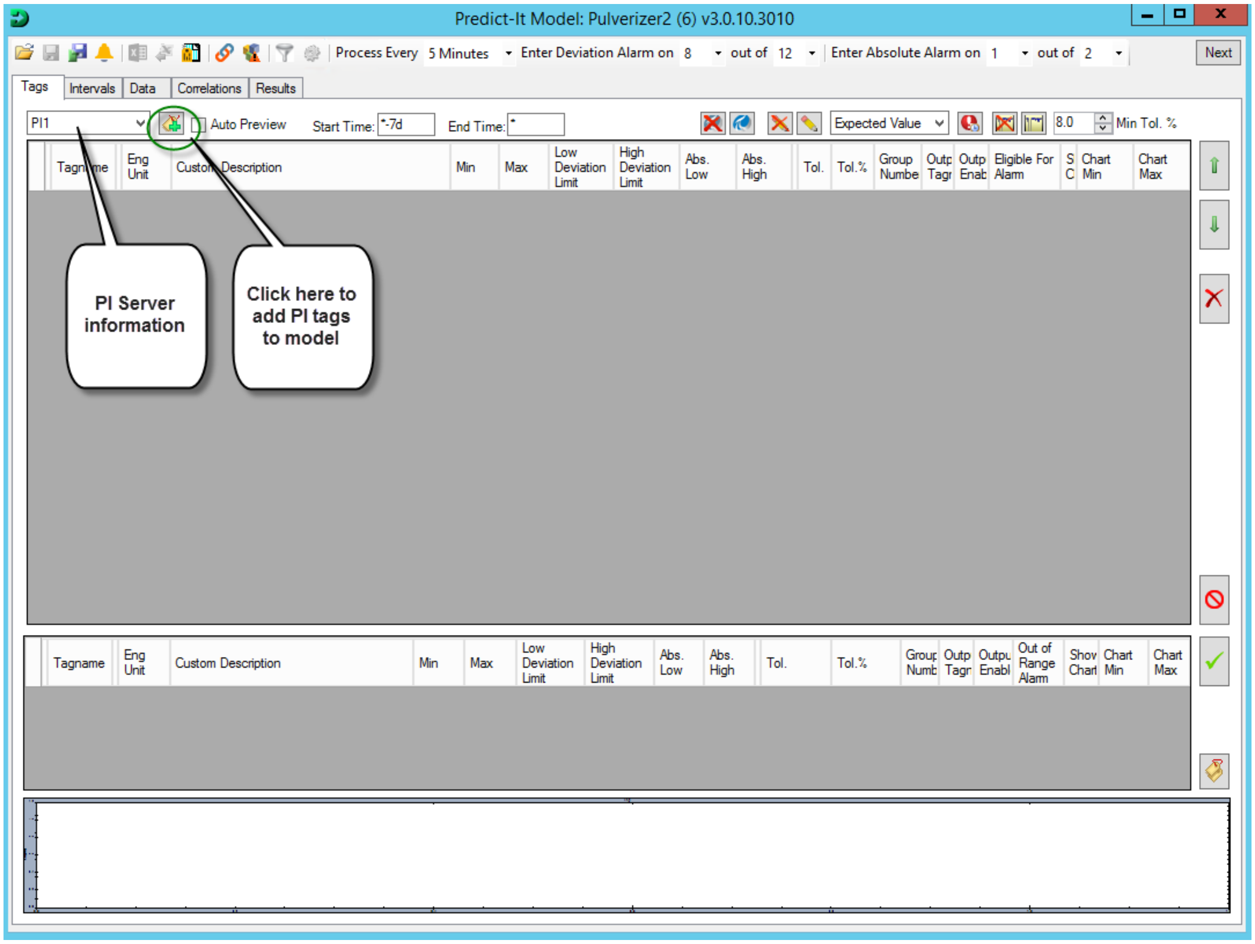
3. Type in a title to name the model. For this example, use "Pulverizer2".

4. Hit the **Create** button next to the Model Name text box. This will open the Model Builder user interface.

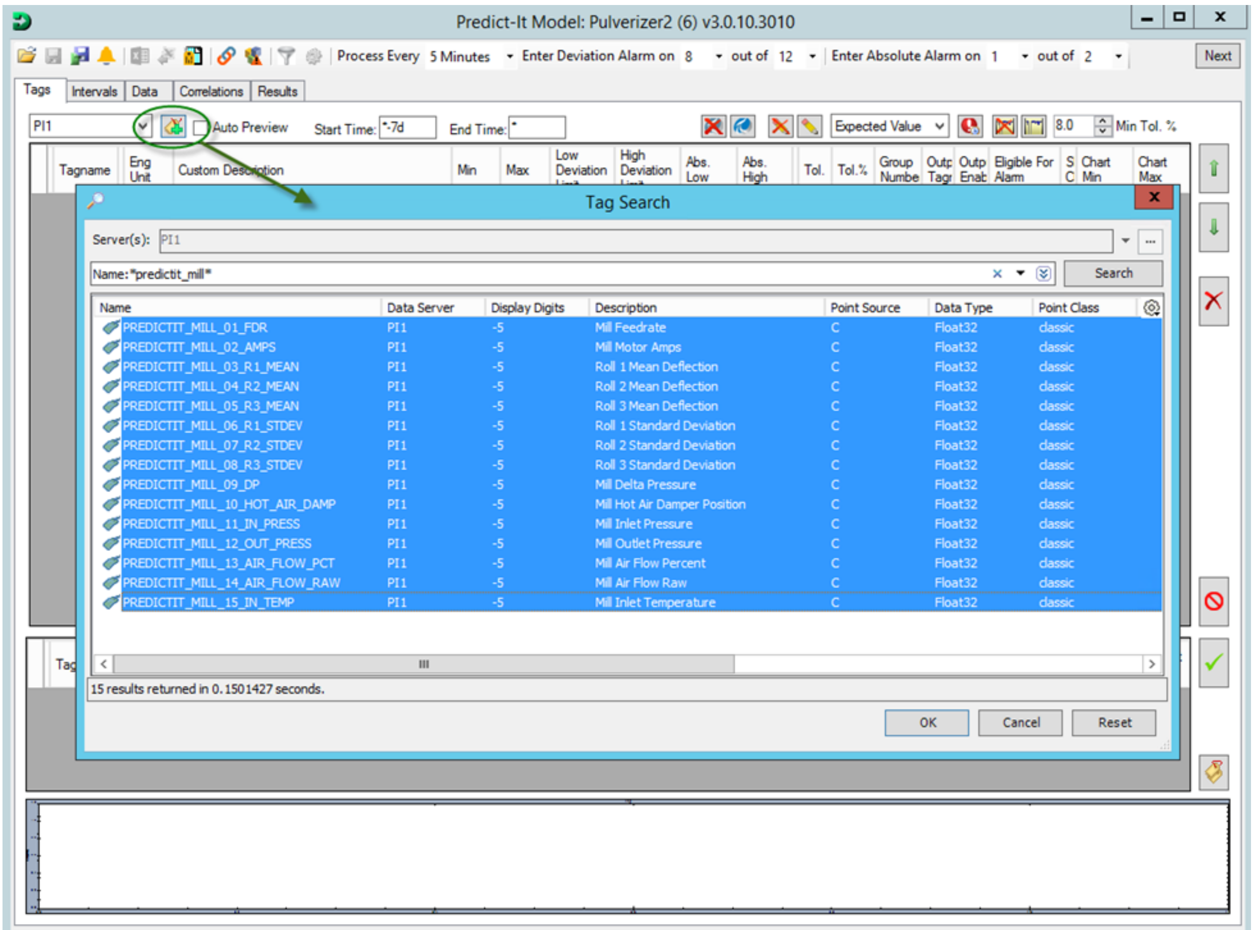


5. In the Model Builder, in the upper left-hand corner, it shows the currently selected PI server i.e. PI1.

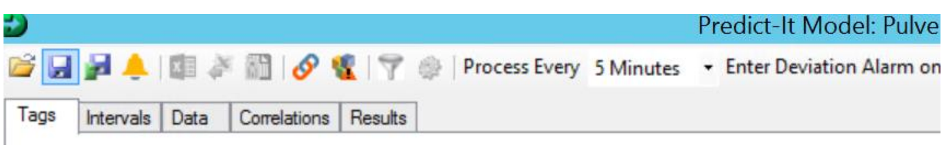
6. Next to the PI server dropdown box is the **Add Tags** button. Click this to add tags to the current model.

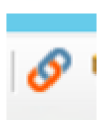


7. In the search box, use "Name:*predictit_mill*. This will retrieve all tags related to the Pulverizer.

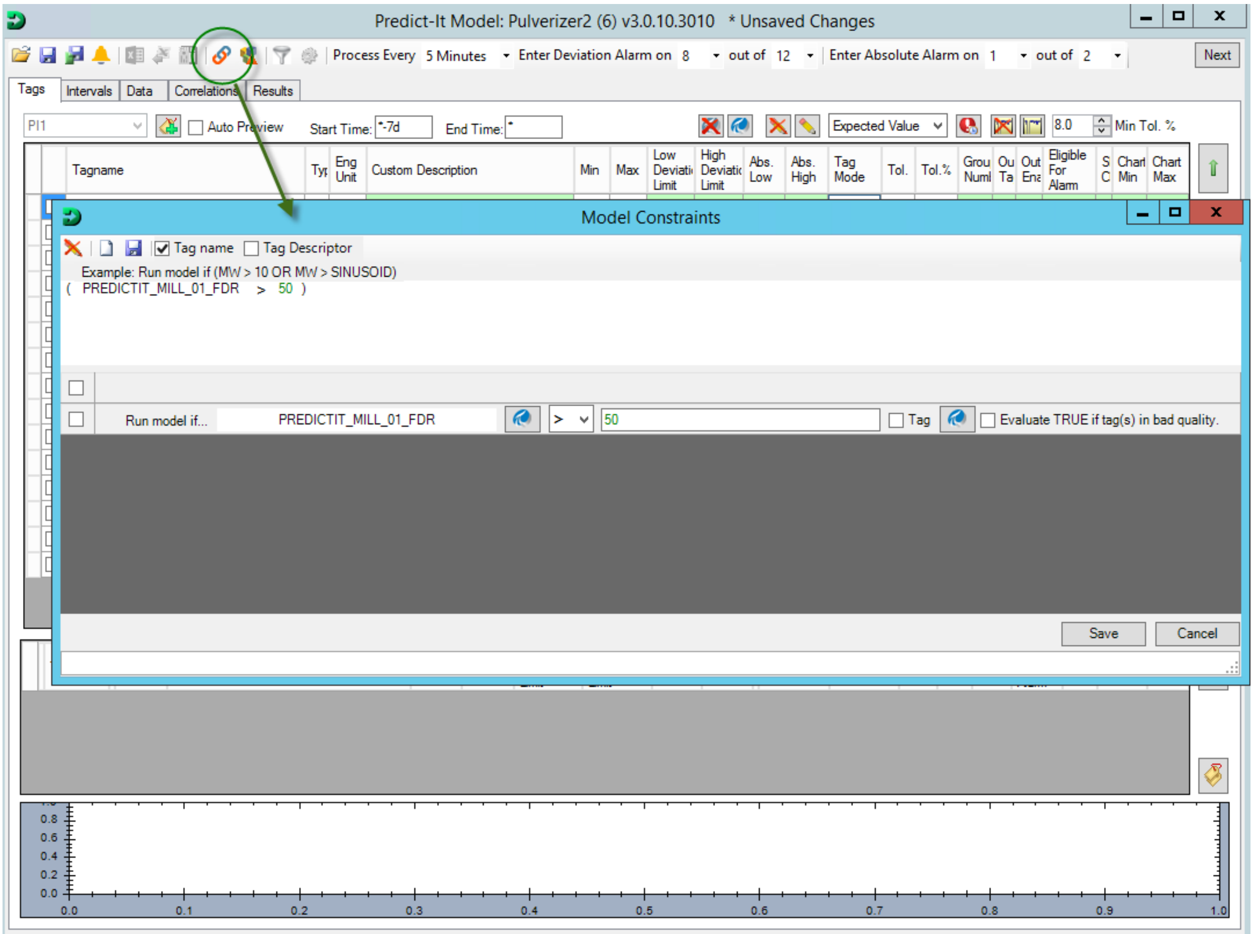


8. Select all and hit **OK**.

9. 

   In the top menu bar, click on  button to open the Constraints screen.

10. Set a constraint such that the model will run only if PREDICTIT_MILL_01_FDR > 50 and hit **Save**.



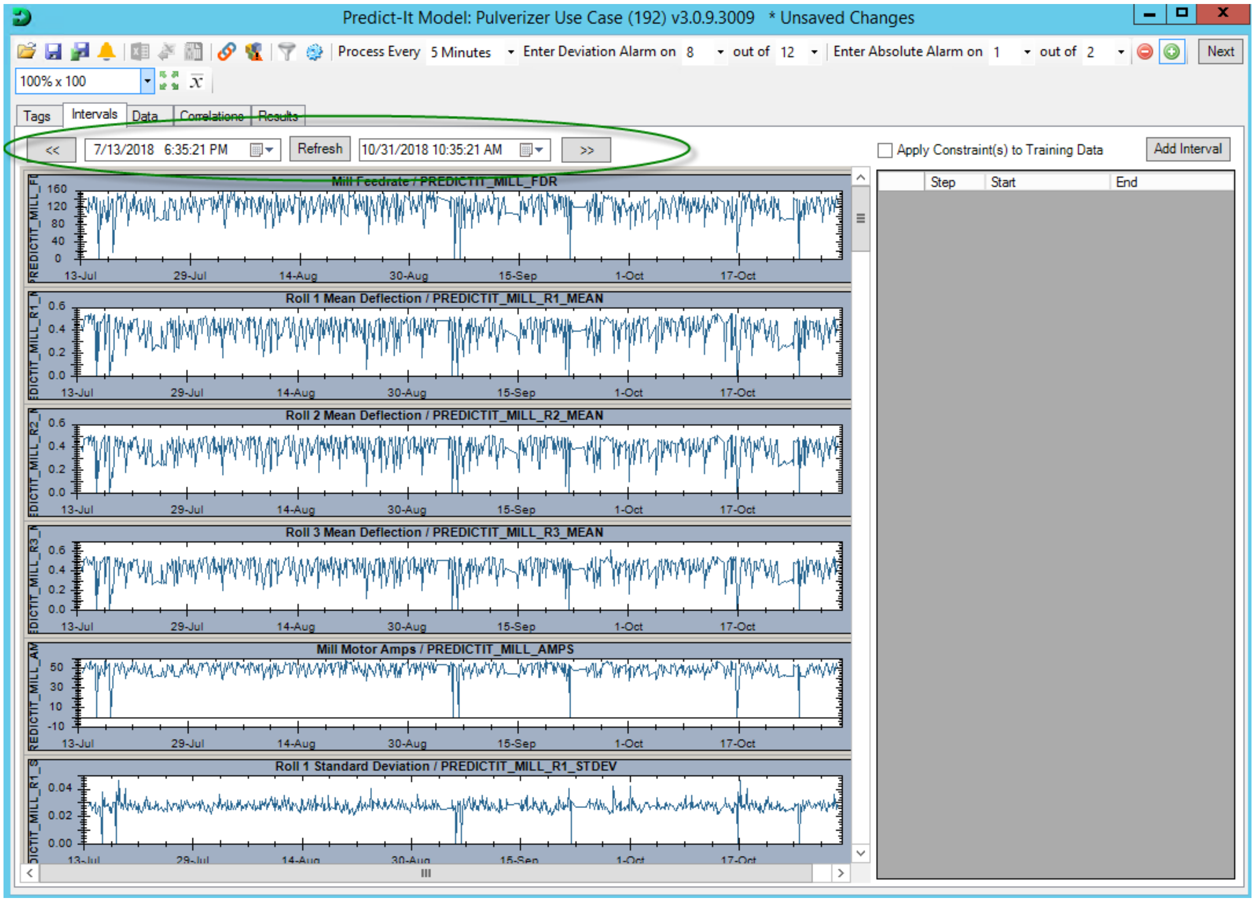
11. You may arrange the tag list in a specific order to suit your preference – perhaps on known relationships among the tags. This makes interpreting the results easier.

    For this example, we will leave the tags in the default order.

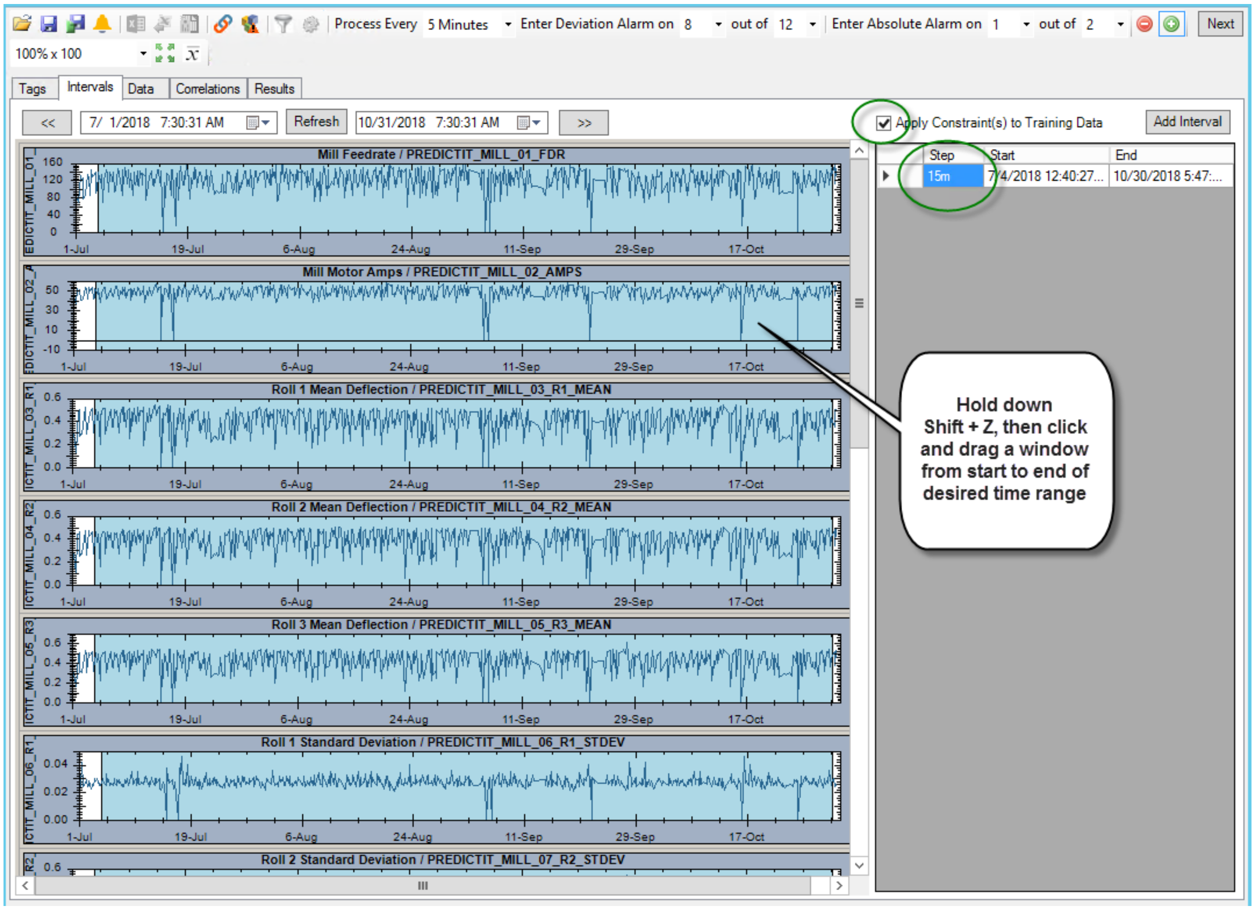
12. Click **Next** in the upper right-hand corner to move to the **Intervals** tab.

Step 3: Train the model

1. Select a time range for training.
   For this exercise, use 07/01/2018 as the start time and 10/31/2018 as the end time.
   Click **Refresh** in the middle of the dates.
   This retrieves 4 months of Pulverizer data for training the model.



2. You can select training data by holding down **Shift +Z**, clicking on the desired starting point, and dragging a selection window to the end point. Hold down Shift +Z, click near the beginning of one of the trends and drag to the end of that trend.
   a. Once a time range is selected, you can also go to the right-hand column and again adjust the Start and End times.
   b. The step can also be adjusted to set how often a data point is collected from the data range. For this example, set the Step to 15m.



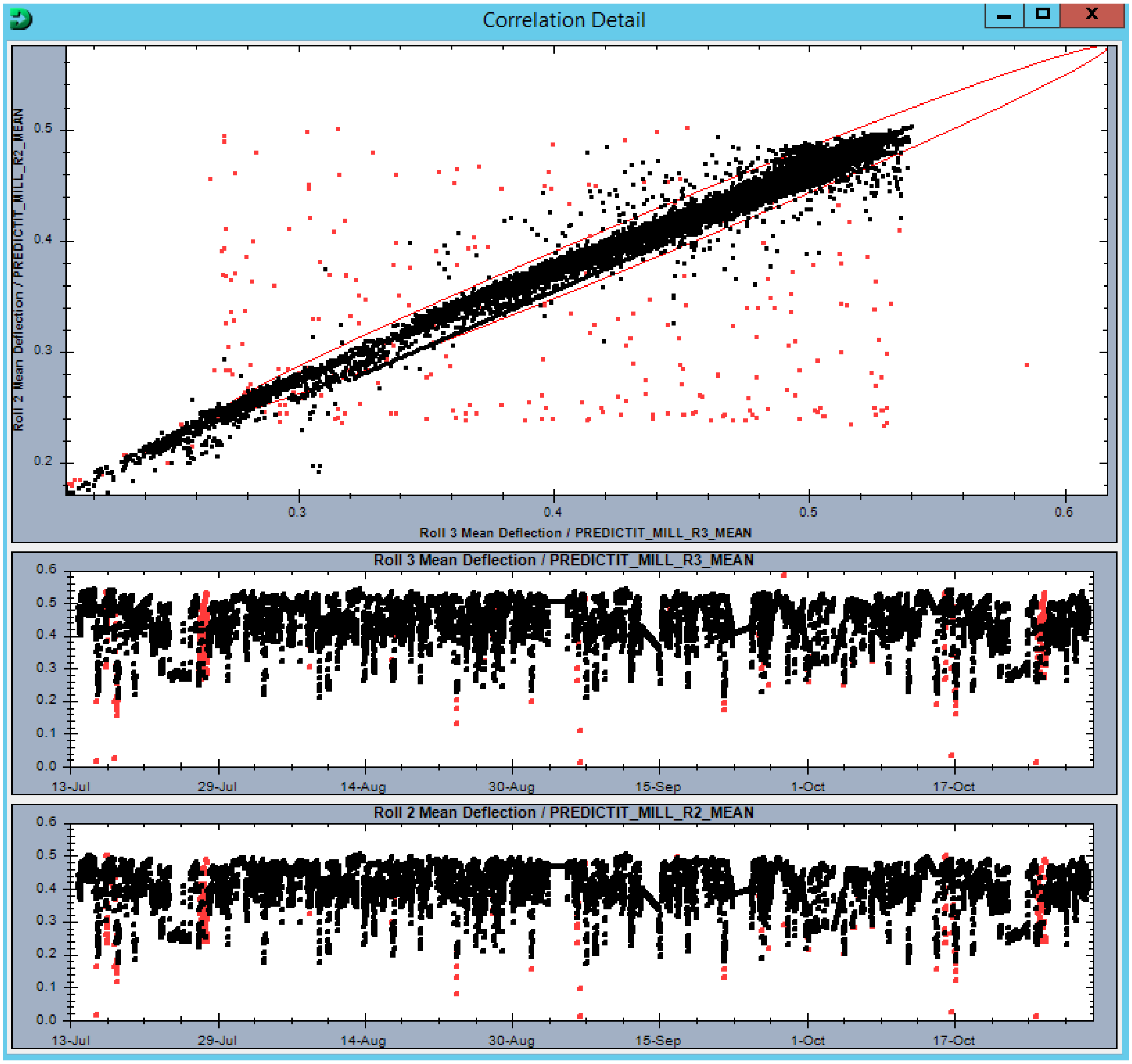
3. Check the "Apply Constraint(s) to Training Data box on the right side. This will exclude any data from the training matrix when the feeder rate is not above the 50 tons/hour specified in the constraint. Click **Next** in the upper right-hand corner.

4. On the **Correlations** tab, click on the **Show All Correlations** button just above the **Results** tab label.

5. Use the **rho (ρ)** button to display the correlation coefficients for each scatter plot.

   The minimum correlation coefficient allowed into the training matrix is set next to the **rho (ρ)** button. Ensure this value is set to 0.20.

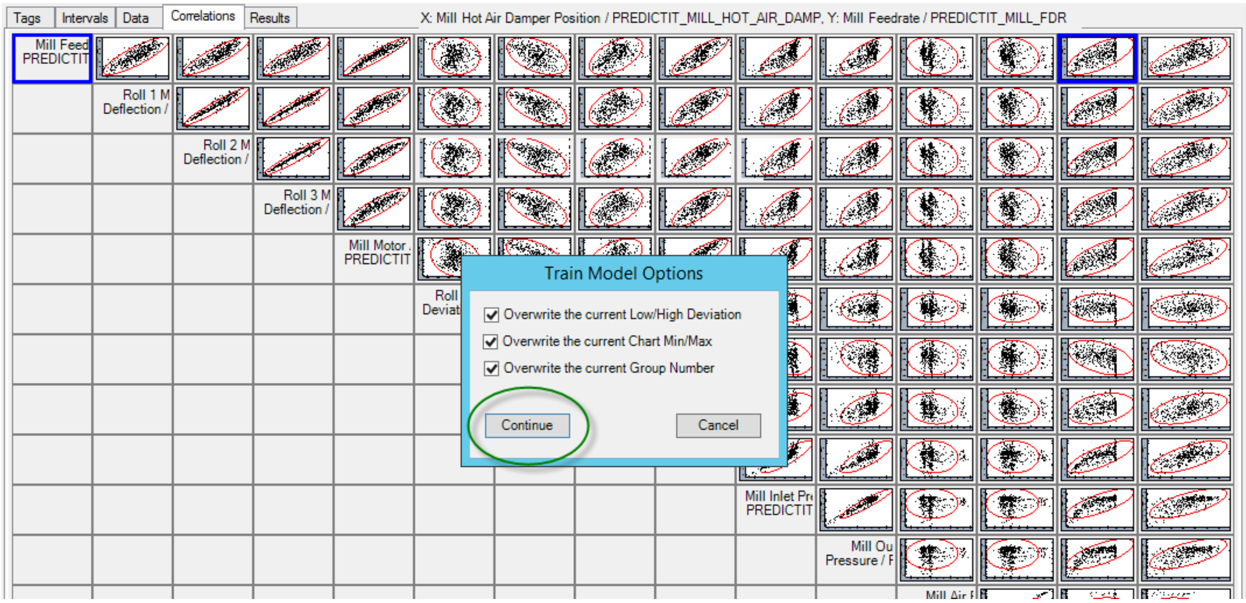
6. Double-click on any X-Y scatter plot to enlarge it.
   The corresponding time plots for the two variables are also displayed.
   You can use any of the three trends to select data to be removed from the training matrix.
   Lasso the points (click mouse-left button and drag to enclose the desired data points), the **Outlier Options** box will pop up. Select **Exclude** to remove data.
   Close the scatter plot when done.



7. For this exercise, we will use the **Multivariate Outliers** function (above **Correlations** tab label) to automatically remove any statistical outliers from the training matrix.
   See image from item 7 above.

8. After the outliers have been removed, the remaining "normal operation" data is used for training the model.
   Click **Next** in the upper right-hand corner. This step prunes the selected data.

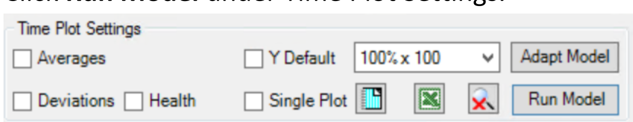
9. Once the data has been pruned, the model is ready to be trained.
   Click **Next** again to train the model, the **Results** tab will automatically come up.

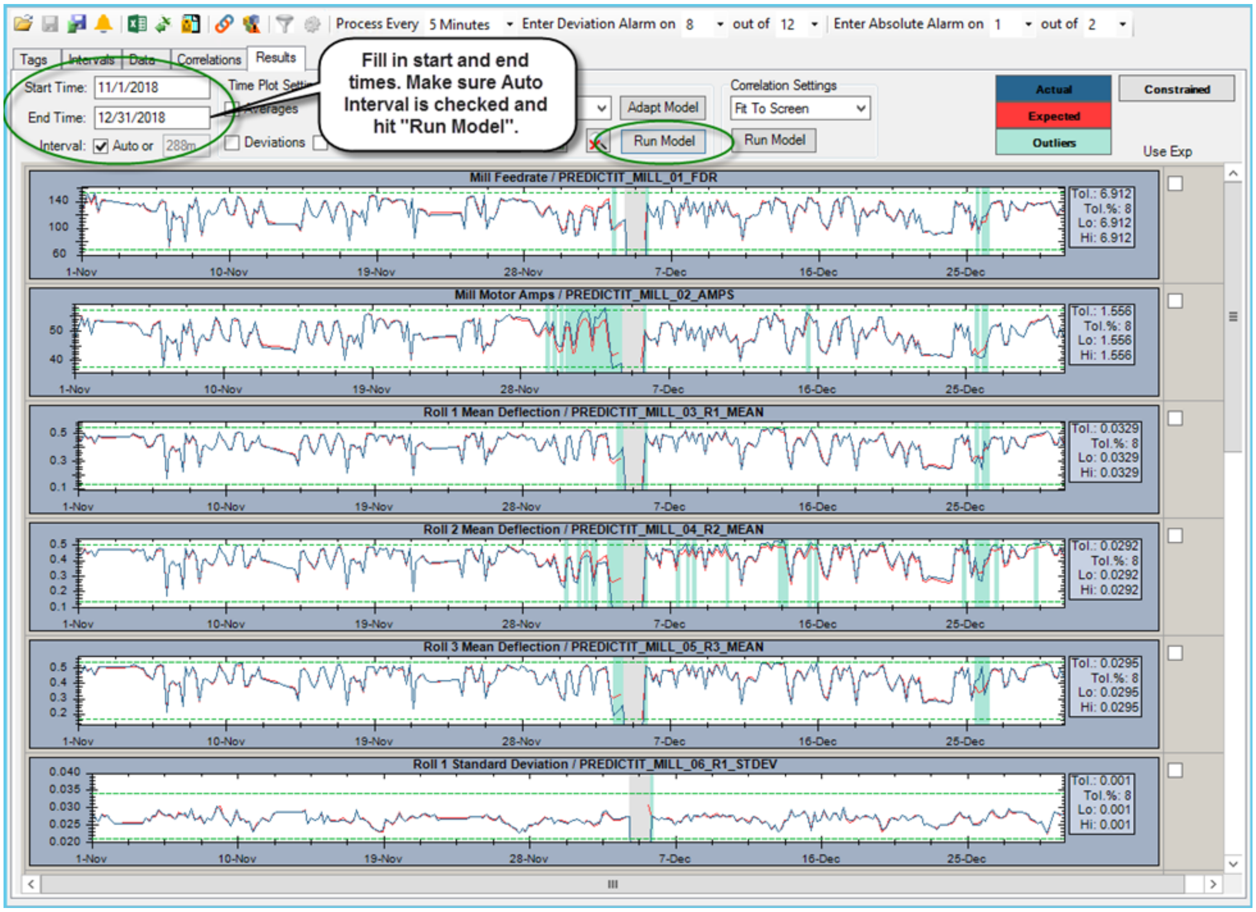


Step 4: Model validation with historic fault data

1. The **Results** tab automatically runs the model for the last 3 days.
   You can adjust the start and end time to run the model against historical data.
   Fill in 11/01/2018 for the Start Time and 12/31/2018 for the End Time.
   Make sure that "Auto" is checked for Interval.

2. Click **Run Model** under Time Plot Settings.

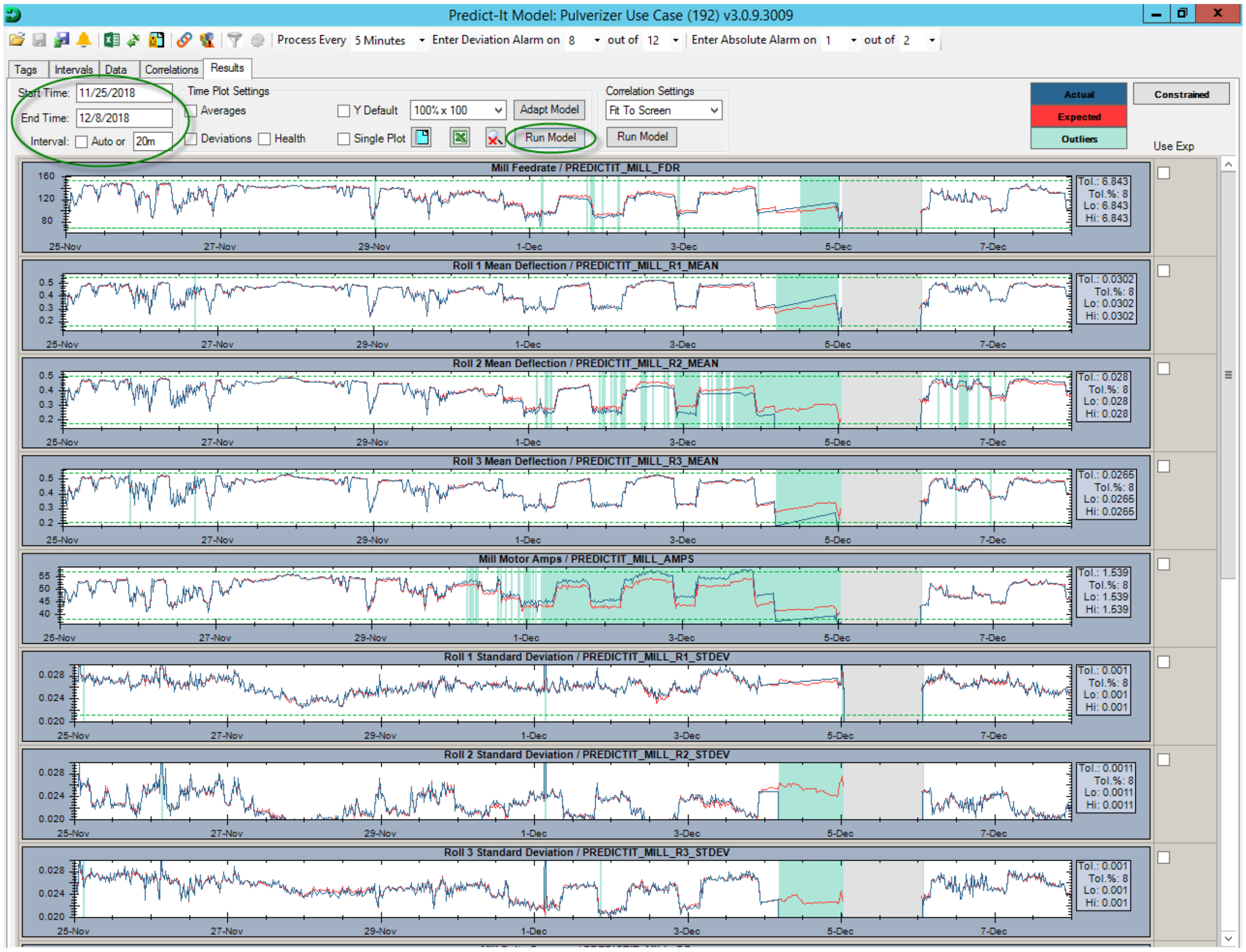




3. You can observe that we can pick out the failure from the trends. Pay attention to variables that could have indicated early warnings of the failure.

4. Focus on the time period of early warning. Change the Start Time to 11/25/2018 and the End Time to 12/08/2018.
   Uncheck "Auto" for Interval and set it to 20m.

5. Click **Run Model** under Time Plot Settings.



6. Note that the Mill Motor Amps started to run higher than expected around Nov 30 and continued to read high up to the point of failure. The Roll 2 Mean Deflection also deviates from the expected value - and is lower than expected beginning around Dec 2.
   Together, they indicate a possible bearing issue with Roll 2.

7. Roll 2 locked up on Dec 4. Note the reactions of other variables to this failure.
   With APR, the first indication of a potential problem is noticeable as early as 5 days prior to the ultimate failure.

Step 5: Model deployment for live data

The model used in Step 4 with historical data is also used for live data – and is active whenever the PredictIT service (windows service) is running. No additional deployment is necessary.

Exercise 4 – Condition assessment rules and asset health score

Objective
Apply the appropriate condition assessment rules to process/equipment measurements and calculate an overall asset health score.

This Exercise is a walk-through; it does not include specific hands-on items

Solution
The solution uses Asset Framework Analytics capabilities to convert a “Raw Value” (PI tag value) to a normalized i.e. a “Case Value” (AF Attribute). And then, by applying a Weight%, it is transformed to a Score.

Thus, each measurement gets a normalized weighted score (0 to 10) by applying a condition assessment rule. And, then the normalized scores are rolled up to arrive at a composite asset health score. The Weight% applied to each attribute depends on its contribution to the asset health.

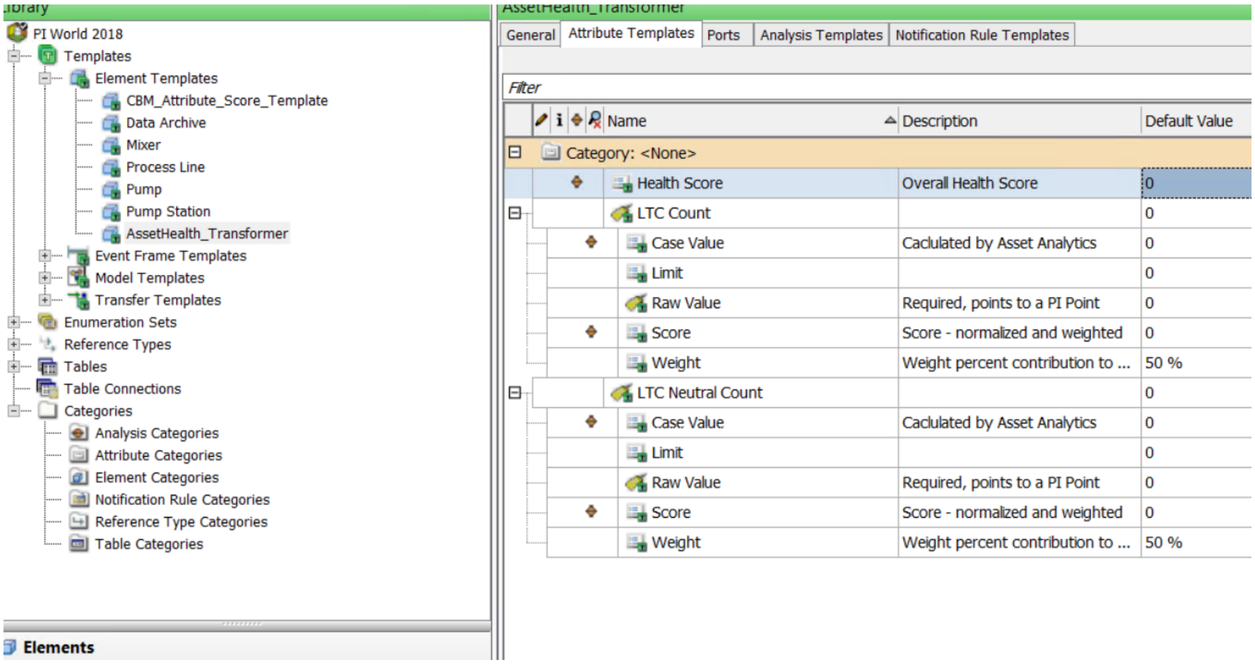
The composite asset health score ranges from 0 to 10 (0=Good, 10=Bad)

Let’s use a Transformer - and consider the following:

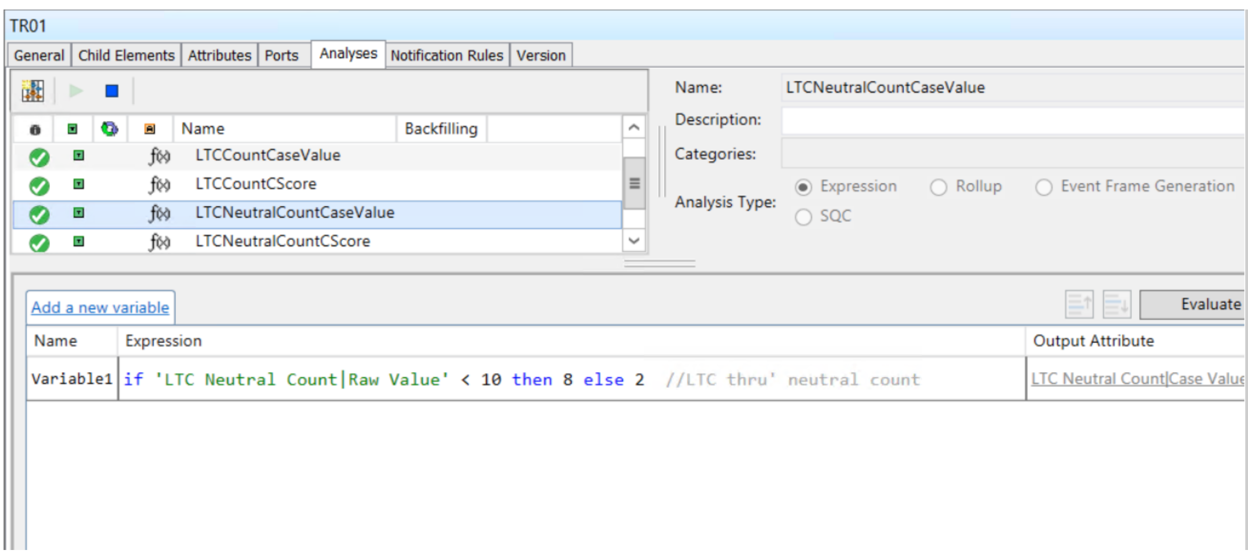
- LTC counter operations (LTC= Load Tap Changer)
- LTC through neutral count
- DGA (dissolved gas analysis) detectable acetylene
- DGA High gas rate of change
- Low dielectric
- High water
- Low nitrogen pressure

For illustration, let’s use the first two, i.e. LTC counter operations and LTC through neutral.

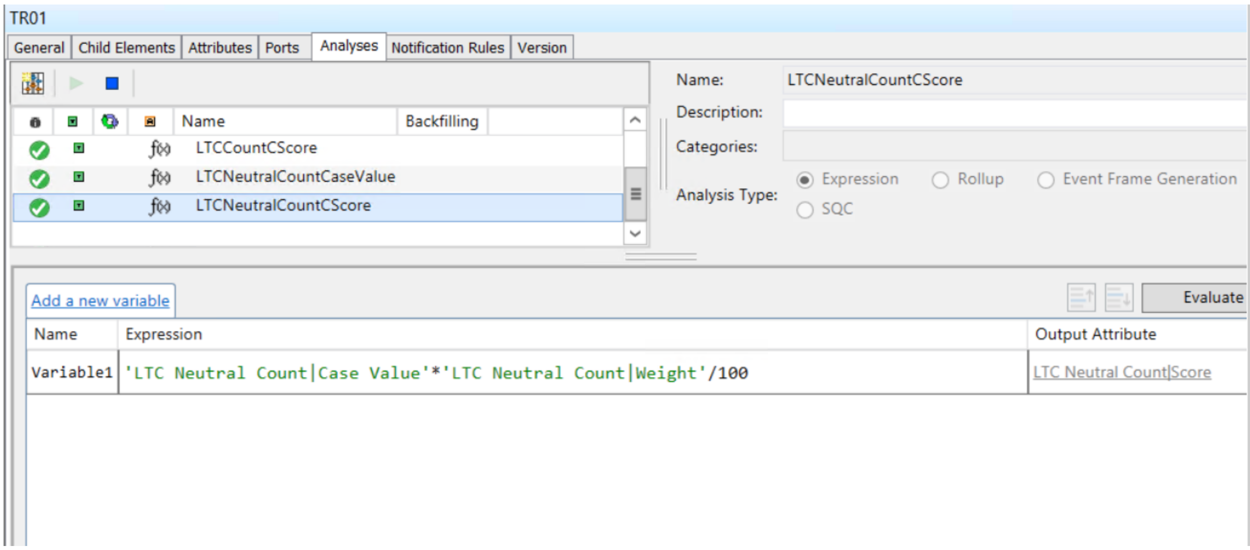
The screen below shows a Transformer template.



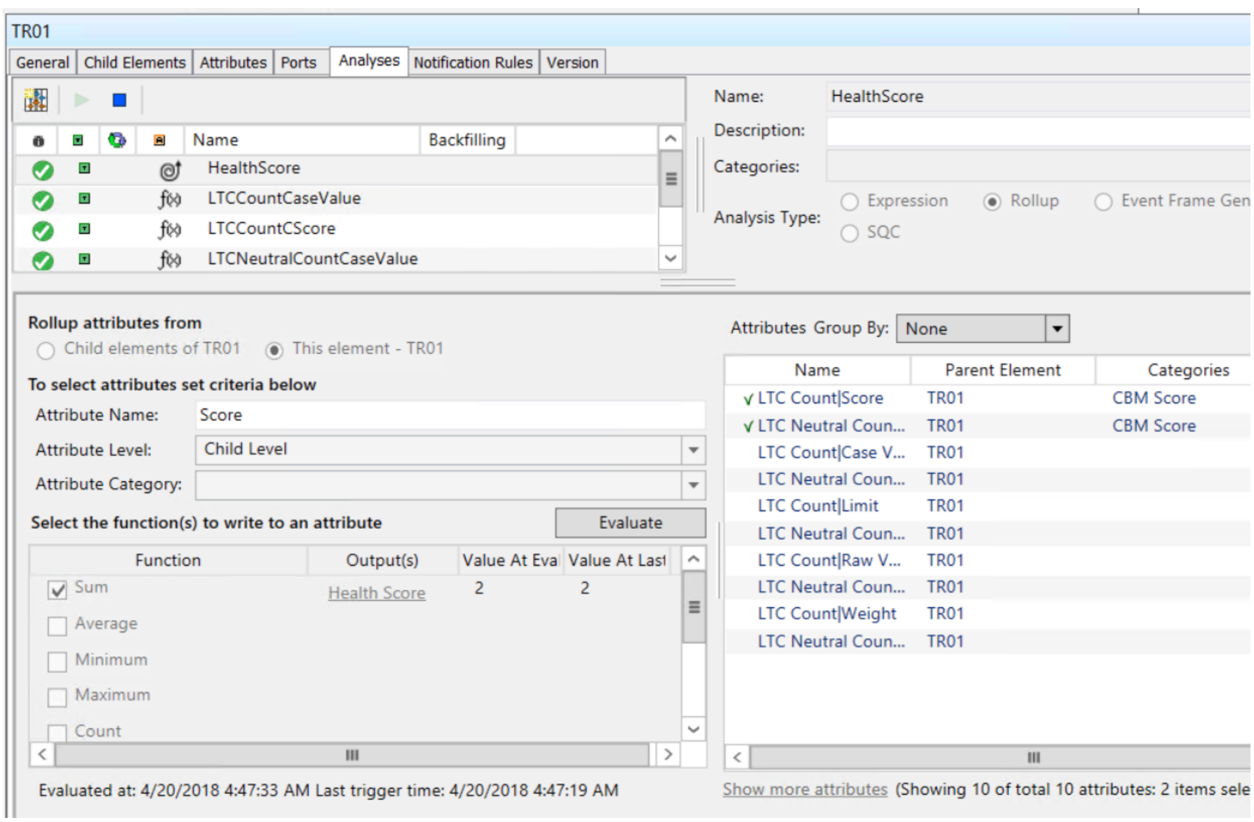
Individual attribute Case value calculation:
Note that the analysis uses “Raw Value” as input and writes to “Case Value”.



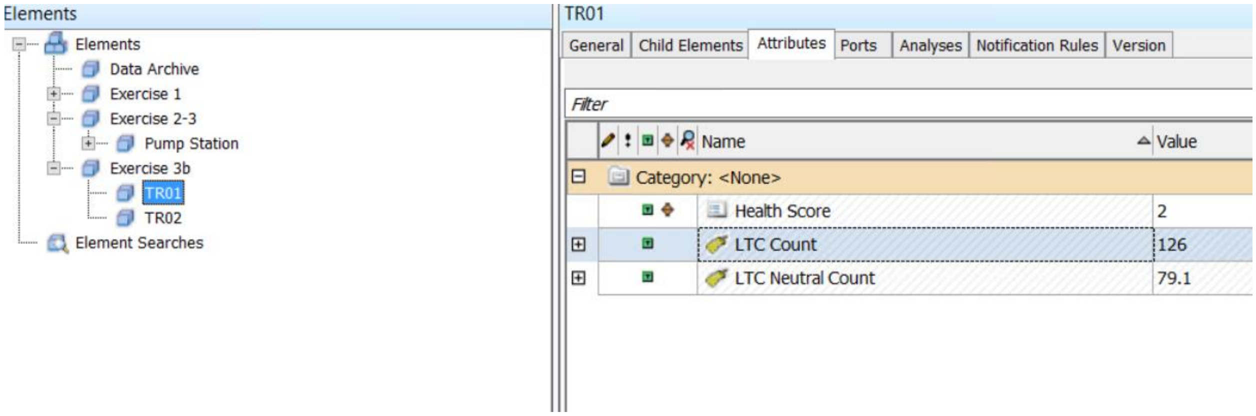
Individual attribute weighted score:



Overall Health Score is a roll up of the individual Attribute scores:



And, as you configure Transformers using these templates, the composite health score is periodically calculated by PI System Asset Analytics.



The composite health score for TR01 is 2 i.e. asset is in good health (0=Good, 10=Bad).


Have an idea how to improve our products?
OSIsoft wants to hear from you!
https://feedback.osisoft.com/

Save the Date!

OSIsoft PI World Users Conference in Gothenburg, Sweden. September 16-19, 2019.

Register your interest now to receive updates and notification early bird registration opening.

https://pages.osisoft.com/UC-EMEA-Q3-19-PIWorldGBG-RegisterYourInterest_RegisterYourInterest-LP.html?_ga=2.20661553.86037572.1539782043-591736536.1533567354