OSIsoft
PI World
SAN FRANCISCO

MARCH 23-26, 2020

PI World 2020 Lab

See the Light, Easy wins to improve your Asset Framework Experience

Operational IntelligenceCopyright
Copyright & Trademark
© Copyright 1995-2020
OSIsoft, LLC
1600 Alvarado Street
San Leandro, CA 94577

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of OSIsoft, LLC.
OSIsoft, the OSIsoft logo and logotype, Managed PI, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, RLINK and RtReports are all trademarks of OSIsoft, LLC.

All other trademarks or trade names used herein are the property of their respective owners.

U.S. GOVERNMENT RIGHTS
Use, duplication or disclosure by the US Government is subject to restrictions set forth in the OSIsoft, LLC license agreement and/or as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

Published: March 16, 2020

Table of Contents

1. Introduction

1.1. Overview of this Lab

This Lab offers you a wealth of best practices that will substantially improve your experience of the PI System. We will focus on **Asset Framework** (a.k.a. **AF**) and **Asset Analytics**, and we will use PI Vision to look at the consequences of adopting good and bad practices in AF modelling. We will work on AF using **PI System Explorer** (a.k.a. **AF Client**).

The Lab is conceived for users that already have some hands-on experience with Asset Framework and Asset Analytics. Should you feel insecure on some aspects of this Lab, please consult the References and Additional resources sections at the end of this workbook.

Some sections are marked by a ***NERD ALERT***! For time constraints, during the Lab we won't cover the topics described in these sections. If you are already familiar with the concepts illustrated in some of the exercises, feel free to go through the NERD ALERT sections yourself.

You should be able to reproduce each step of this Lab by following this workbook. Should you get stuck or want additional clarifications, please do not hesitate to ask questions to the Lab instructors and helpers.

1.2. Why adopt Asset Framework for your enterprise?

The introduction of Asset Framework in 2008 has marked the passage from offering a "simple" data historian (the Data Archive) to providing a **full data infrastructure**, which can be used at all levels of an organization to access data in an intuitive, user-friendly way.

A data infrastructure extends the scope of your PI System way beyond site-level data historians, by (1):

- creating a universal, master data system. An infrastructure simplifies governance so important data can be formally managed. It also supports processes that democratize data so people in any organizational department or at any level have real-time access to data and information
- removing the work associated with finding, converting and organizing data. People no longer have to wrestle with complex data access chains to find the "right" data for analysis or reporting
- making data available for multiple purposes. Instead of being used for preordained purposes, data can be accessed and shared with people and systems to support systematic enterprise control and communication throughout its parts

- reducing overall complexity and cost. An infrastructure eliminates the skilled resources, customized solutions and coding associated with connecting data from isolated point solutions, applications and historians
- creating enterprise readiness. Enterprises can rapidly take advantage of advances in IT applications, technology and solutions without having to re-integrate or rip and replace enterprise operational technologies (OT) architectures.

With its 10+ years of history, Asset Framework has become a crucial component of the PI System for its capacity of structuring operational data from the Data Archive and integrate this information with contextual metadata.

In combination with other components of the PI System (and specifically Asset Analytics and Notifications), Asset Framework is being used by our customers for a variety of purposes, such as:

- condition-based maintenance
- asset-health real-time monitoring and anomaly detection
- performance optimization
- diagnosis of downtime events
- data modelling and data cleansing for integration with third-party advanced analytics

It is important to mention that the successful deployment of AF across the enterprise depends on an initial investment in brainpower. Asset Framework is not a turn-key solution because it aims at modelling any industrial process, for a large array of business needs. When you start modelling in AF, you need a strategy to make sure that you quickly derive business value from it. What are the possible approaches?

1.3. Possible modelling approaches

If you've worked with the PI System before, most likely you are already familiar with what a Data Archive and a PI Point are. If not, here is a brief definition of both:

Data Archive: the component of the PI Server that stores and archives time-series data and serves it in real time throughout the PI System and your information infrastructure.

PI Point (a.k.a. **PI Tag**): the **basic building block of a PI System**. A PI Point represents a single data stream coming from a data source and being tracked by the PI System. Examples of PI Points could be the temperature of a bearing ring, the state of a digital alarm or the result of a KPI calculation.

1.3.1. Bottom-up approach

When a company's data management team (meaning, the PI administrators and possibly a few *power users*) decides to adopt and deploy Asset Framework, normally it's been using Data Archive for a while and it has created a certain amount of PI Points. Since the PI Points already exist, one approach to AF modelling would be to group all PI Points by asset, and then attempt to group the assets by similar PI Points. These "similar-object groups" become AF element templates (more about AF templates in Section 2.1), and the "objects" become AF elements.

Typically, the following step is to layer static metadata (such as the asset type, its installation date, the date of last maintenance and so on), KPI calculations and other calculations needed for real-time notifications.

This **bottom-up approach** has the advantage of being a relatively clear, reality-based path to follow.

We know the PI Points, and we know we can layer analyses on top of them… but where's the vision? Why are we doing any of this? And what implies that the result we happen to create will be at all useful?

When we start with what we have instead of what we want – we don't know what we are missing (what other data sources could be valuable?), and we come to the problem with a **mindset dominated by the instrumentation**, which may not be appropriate for process-decision support.

Also, we might quickly abandon the task as we get overwhelmed by the sheer number of PI points that we want to group by similar assets.

1.3.2. Top-down approach

Alternatively, some users start modelling AF with a focus on the enterprise hierarchy – they put the corporate name at the top, then build a logical tree and at the bottom of the tree they attach the individual equipment units. The good news is that this **top-down approach** focuses on a goal – building a data infrastructure.

The less than good news is that the requirement – the expectation of value is still not that well defined. For some customers **this process of sorting into the hierarchy can take a long time** and, in the meantime, the "incomplete" hierarchy is not that useful, even as a data infrastructure.

This approach may postpone the business value until the data infrastructure is “complete”; this can result in a long wait with delayed value, cutting corners to meet deadlines, etc. Also, creating a very ramified hierarchy in AF, with e.g. 8 levels to drill down to get to the equipment unit of interest, would result in a lot of clicks and a frustrating end-user experience (we’ll come back to this while working in PI Vision).

1.3.3. Value-oriented approach

The approach we recommend is the **value-oriented approach**. Whatever work you were doing **before** you started to use Asset Framework – we encourage you to keep working with the same goal in mind and not expect to postpone the business outcome for very long.

For example, we could build an analysis directly into Asset Framework (using Asset Analytics) instead of creating it in many different Excel spreadsheets using Excel and PI DataLink functions. This will help you create a centralized, consistent version of the truth on the AF Server, while PI DataLink will be mainly used to present the final results (i.e. mainly as a visualization tool, and not as a calculation tool).

Also, we can templatize AF elements and analyses with the goal of building reusable PI Vision and PI ProcessBook displays rather than multiple, tag-based displays.

The idea here is to initially build our AF templates with a simple, clear project in mind, and expand the scope of our Asset Framework project as we knead new requests into the mixture and identify new value opportunities.

Ultimately – implementing Asset Framework is not so much a project as a paradigm shift. It’s not so much “bottom up” or “top down” but how to reconcile end-user efforts with internal company standards. It’s reconciling the ultimate data infrastructure with the needs of value-giving analyses and display projects; reconciling the strategic with the tactical.

2. Building your Asset Framework database

This Chapter tackles many of the most common questions that you might encounter when modelling your AF Server. Section 2.1 highlights the importance of using AF templates, Section 2.2 presents the use case created for this Lab, Section 2.3 guides you in navigating PI Vision displays in order to identify a few issues that we solve step by step in Section 2.4. Each exercise focuses on a specific aspect of AF administration.

2.1. Why AF element templates

How long does it take to deploy a new asset and integrate it with existing displays?

How easily can you tell whether the problem is due to a single bad actor, or if there is a larger pervasive issue with a certain equipment manufacturer or region?

The ability to reliably bring new units online or run asset-to-asset comparisons relies on standardization. These standards should be simple to implement, deploy, govern, and update from a central location. With Asset Framework, you can easily create a view of your asset with associated metadata that integrates and contextualizes data from multiple repositories, whether from the Data Archive or other business or maintenance systems. These asset associations can then be turned into a template that applies to other similar assets, so that every boiler, every pump, or every transformer has a reliable, consistent view. Leveraging templates means that when you create a new efficiency calculation, that same **calculation** can be **applied automatically, consistently to all units with the same template**. If new sensors or a different maintenance database is added, these updates can be managed centrally, thereby **reducing administration time and errors**. (2)

AF templates should be used even when the template has no attributes in the beginning. For example, you could initially build a template for all the elements representing a site. Further on in the construction of your AF database, you might decide to perform some rollup calculations at the site level (see for example the exercise in Section 3.2.3). Having a template ready for this will make maintenance easier.

For whom is already familiar with other PI software components, here are some practical advantages when using AF templates in the PI System:

- build **templatized analytics** (YouTube tutorial).
- **reuse PI Vision displays** for similar assets (LiveLibrary and YouTube tutorial)
- use **Collections in PI Vision** displays (LiveLibrary and YouTube tutorial)
- use **smart hyperlinks in PI Vision** displays (LiveLibrary and YouTube tutorial)

- build **PI ProcessBook Element Relative Displays** (LiveLibrary and YouTube tutorial)
- reuse **PI DataLink reports** for similar assets (LiveLibrary and YouTube tutorial)
- use the **PI Integrator for Business Analytics** to filter, cleanse and publish data to external databases, advanced analytics platforms and data lakes (LiveLibrary and YouTube tutorial)
- use more efficiently PI Developer Technologies such as **PI AF SDK, PI Web API, PI SQL Client** and **PI OLEDB Enterprise** (general overview here, Hands-on Lab presentation and PI SQL Online Course)

Last but not least, using AF templates greatly improves the **overall performance of the AF Server**. Using templates reduces data duplication in the backend SQL database and drastically improves the performance of searches in client applications.

If you are not familiar with some of the terms or products above, feel free to click on the linked pages or to speak with the instructors and other OSIsoft employees at the end of the Lab.

2.2. Our case study – a sugar and salt packaging company

The packaging company Sweet & Savory Corp has two core products: **bags of sugar and bags of salt.**

The company uses **FORM, FILL and SEAL** (**FFS** from now on) equipment units to produce the bags.

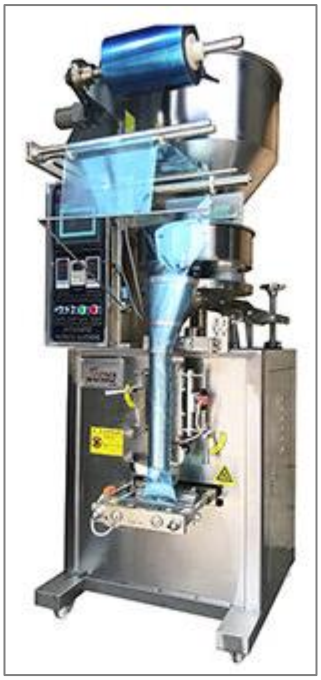


Figure 1. Form, Fill and Seal (FFS) unit (source: **(3)**)

In principle, an FFS machine is a relatively simple unit and it is extensively used in packaging lines.

The material that is to be packaged (in our case salt and sugar) is loaded into a hopper above the machine. A controlled amount is let down into a loading cartridge situated just above the package opening.

The packaging material (normally plastic) is loaded into the machine as a roll and **formed** into a pocket by a series of rollers. The loading cartridge will open above the package opening and deposit the material into a pocket, **filling** it.

A heated sealer will then clamp the plastic above the pocket, **sealing** the package just before a knife cuts the pocket free to drop onto a conveyor where it is taken off for palletizing. This process is then repeated until either the hopper is empty, or the machine runs out of packaging material.

If you'd like to see an FFS unit in action, feel free to watch this YouTube video: [1KG flour Vertical FFS bag packing machine automatic with auger filler granule filling packaging]().

The FFS units used by Sweet & Savory Corp come in two models: ADCO and MF TECNO. The models present similar technical specifications but are manufactured by two different suppliers.

The **chief maintenance engineer** of Sweet & Savory Corp is working on a large-scale condition-based maintenance program and needs to standardize the modelling and monitoring of the FFS units.

The project will be initially deployed in one production site, and if the outcome is positive, it will be deployed at an enterprise-scale level across the globe.

The units should be organized by:

- Supplier (in order to facilitate the filing of work orders)
- Type of product (in order to monitor production and fine-tune the machines' settings)

Also, the chief maintenance engineer would like to integrate the PI System with maintenance information coming from an external table hosted in a SQL Server. The information contained in the external table is reported in Table 1.

Table 1. Maintenance information for each FFS unit

Asset	Model	Serial Number	Installation Date	Last Service Date	Service Crew
FFS01	ADCO	ADA5FFS01	2017-05-01	2019-11-01	Red
…	…	…	…	…	…

For his purposes, the chief maintenance engineer would like to retrieve the **Last Service Date** for the site in question.

The same AF database is also accessed by the **chief process engineer**, who wants to monitor:

- the **number of bags** packed by each FFS unit **per minute**
- the **number of bags** packed by each FFS unit **since the beginning of the day**
- the **moisture content** (which should be kept below 0.1%)
- the **grain size**. This is required only for the sugar bags
- the **bags size** (which can be 220, 500 or 1000 g)
- the **Overall Equipment Effectiveness** (**OEE**) of the machines. The Overall Equipment Effectiveness (OEE) is calculated as: OEE = Availability x Performance x Quality

In addition, he would like to keep track of the **operator** working in the production line on every shift for semesterly performance evaluation.

The data management team of Sweet & Savory Corp has already been working on the company's AF database, and at the moment around 20 FFS units are modelled are monitored.

The team is actively working in a **sandbox environment** to test some calculations and configuration aspects of six FFS units (FFS01 to FFS06).

For simplicity and ease of navigation during the Lab, the sandbox environment is here modelled as a branch of the production AF database. In the real world, the sandbox environment would reside on a dedicated AF database, often hosted on a dedicated AF server.

The exercises presented in this Chapter will start and end with the end-user experience in PI Vision, our web-based visualization tool. PI Vision will give us clear indications on whether something has gone wrong in the AF modelling. Specifically, **the sandbox environment is currently affected by a few issues**, which we want to first identify and then address in Section 2.4.

2.3. Asset monitoring in PI Vision - Directed activities

Step 1. Open PI Vision from the shortcut link on the desktop of the virtual machine. **The first time you open the page it might take a while!**

PI VISION

Step 2. Once on the main page, open the display **Sandbox_Overview_FFS** (Figure 2).

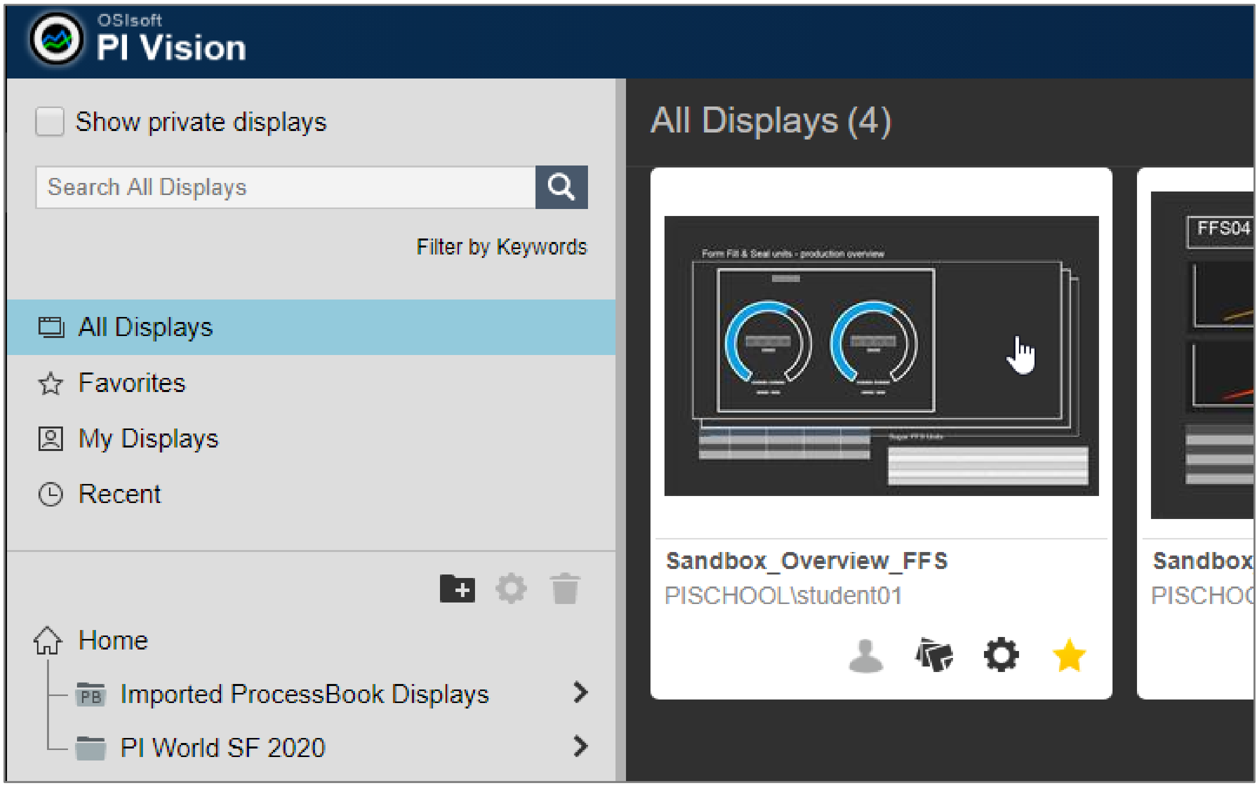
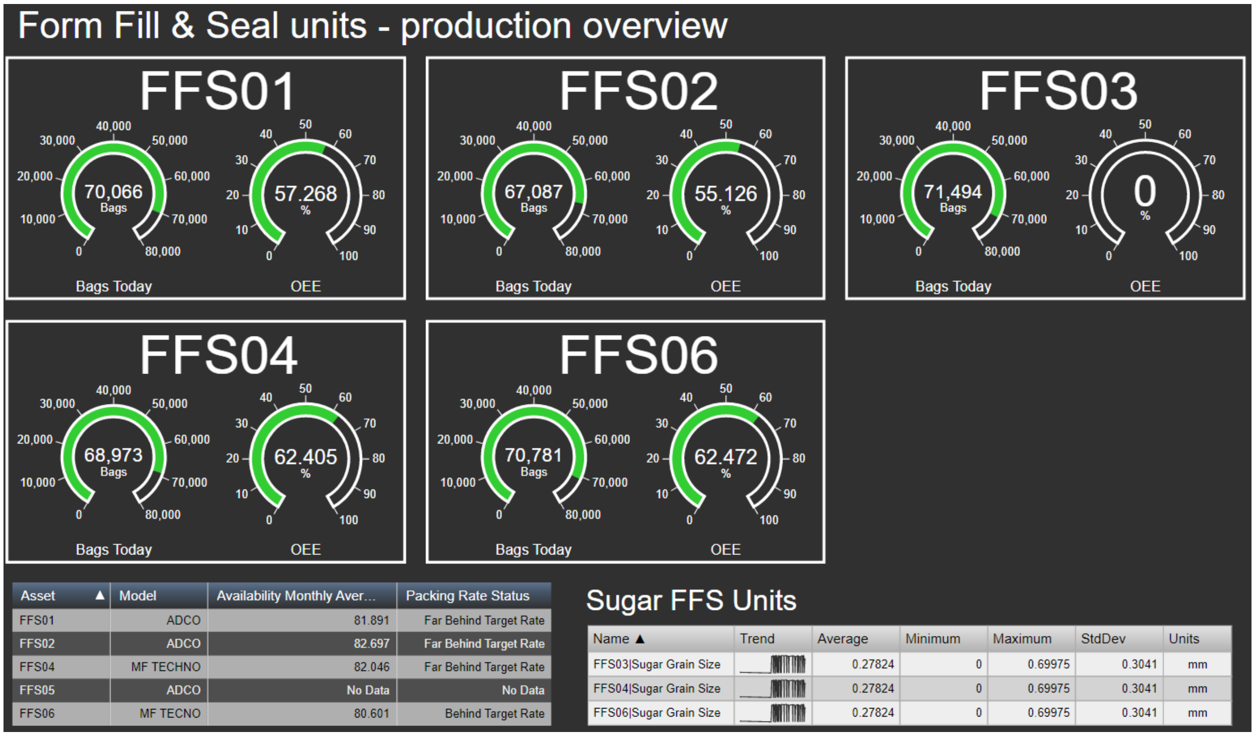


Figure 2. PI Vision main page

This display (Figure 3) provides an overview of the current production from all the FFS units. We can verify which units are underperforming given the daily packaging target, and which units are reporting a low OEE (Overall Equipment Effectiveness).



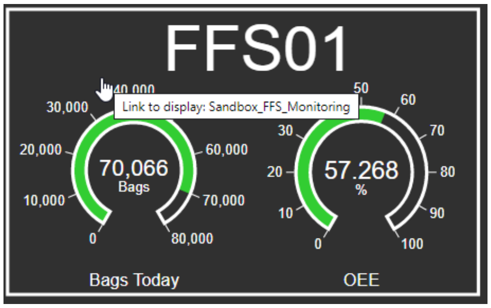
Asset ▲	Model	Availability Monthly Aver...	Packing Rate Status
FFS01	ADCO	81.891	Far Behind Target Rate
FFS02	ADCO	82.697	Far Behind Target Rate
FFS04	MF TECHNO	82.046	Far Behind Target Rate
FFS05	ADCO	No Data	No Data
FFS06	MF TECNO	80.601	Behind Target Rate

Sugar FFS Units

Name ▲	Trend	Average	Minimum	Maximum	StdDev	Units
FFS03|Sugar Grain Size		0.27824	0	0.69975	0.3041	mm
FFS04|Sugar Grain Size		0.27824	0	0.69975	0.3041	mm
FFS06|Sugar Grain Size		0.27824	0	0.69975	0.3041	mm

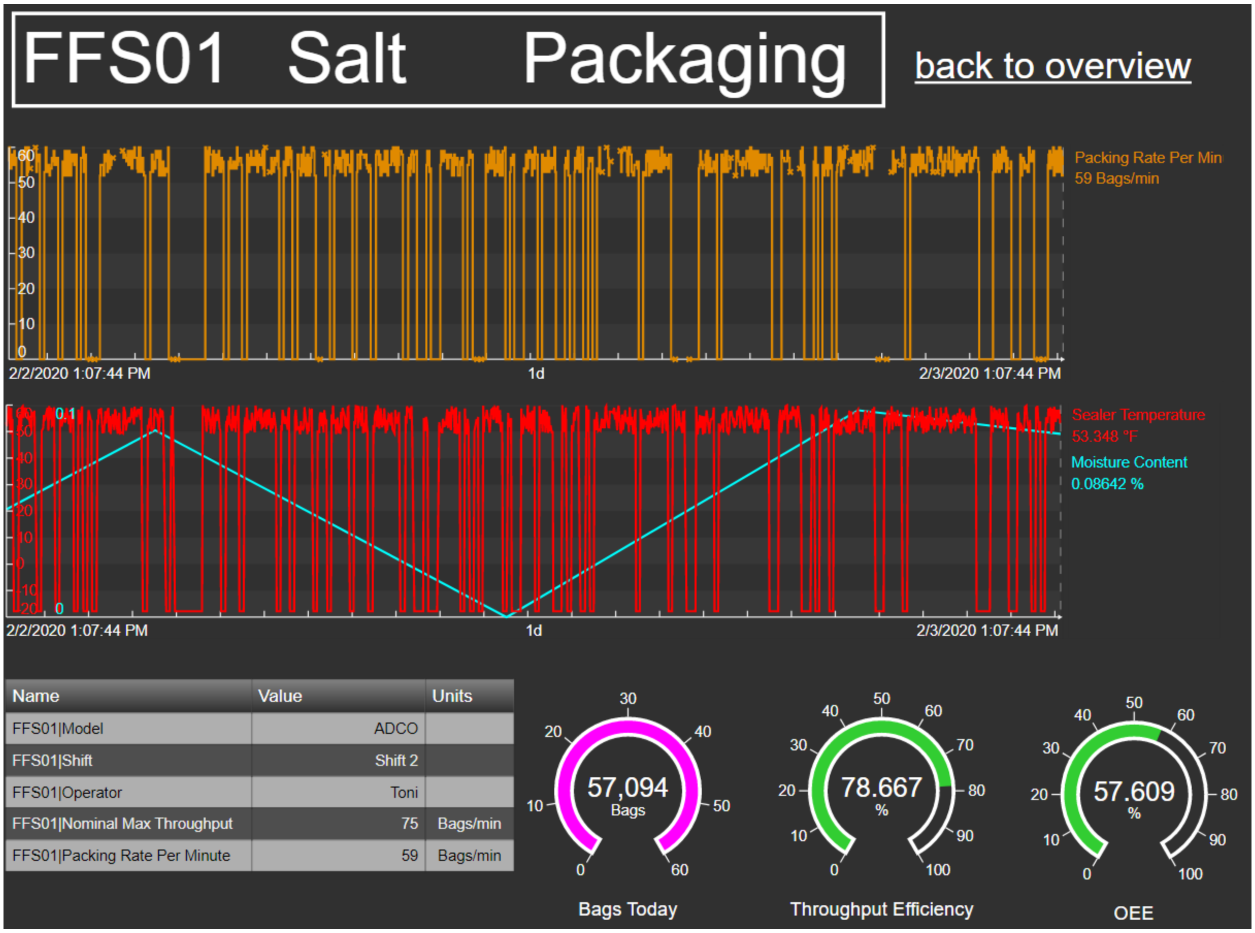
Figure 3. Sandbox_Overview_FFS display

Step 3. Click anywhere in the tile showing **FFS01**:



This will open a more detailed display, **Sandbox_FFS_Monitoring** (Figure 4).

The display reports the trends of the main process variables associated with FFS01, such as the Bags Per Minute, the Sealer Temperature, the Moisture Content etc. The display also offers contextual metadata such as the model, the last service date, the bag size etc.



Name	Value	Units
FFS01|Model	ADCO	
FFS01|Shift	Shift 2	
FFS01|Operator	Toni	
FFS01|Nominal Max Throughput	75	Bags/min
FFS01|Packing Rate Per Minute	59	Bags/min

Figure 4. Sandbox_FFS_Monitoring display

Instead of creating individual displays for each and every asset, PI Vision allows you to reuse a display for all assets presenting similar characteristics, or in PI jargon: "*all AF elements using the same AF template*". Use the **Asset Dropdown Menu** to switch asset and see the display update with the relevant information.

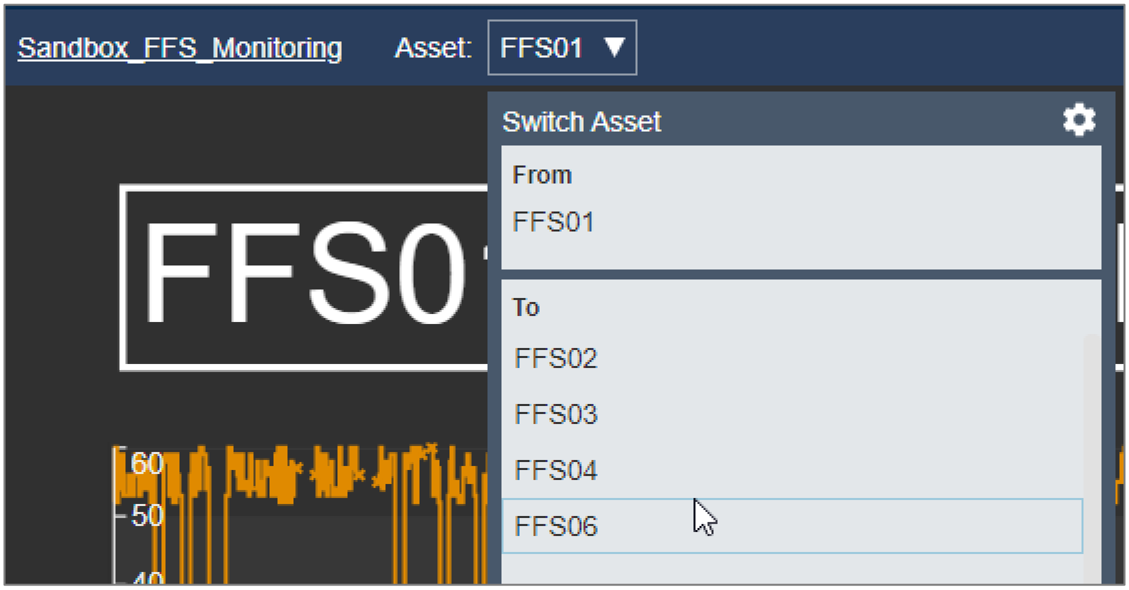
Step 4. Choose **FFS02** and verify that the trends update to reflect the chosen asset.

We know that the sandbox environment includes FFS units from 01 to 06.

Step 5. Try switching to **FFS05** using again the Asset Dropdown Menu. Is this asset available?

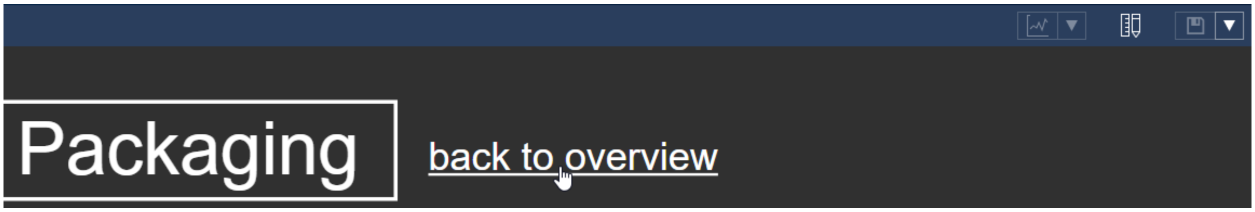
We identified the first issue to tackle!

Issue # 1. Asset FFS05 is not available in the PI Vision Asset Dropdown Menu.



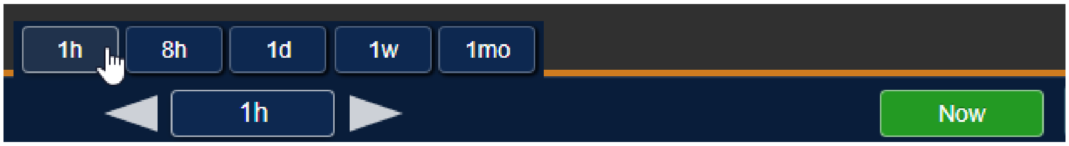
Let's further explore our displays to look for additional issues.

Step 6. Go back to the display **Sandbox_Overview_FFS** using the "**back to overview**" button.



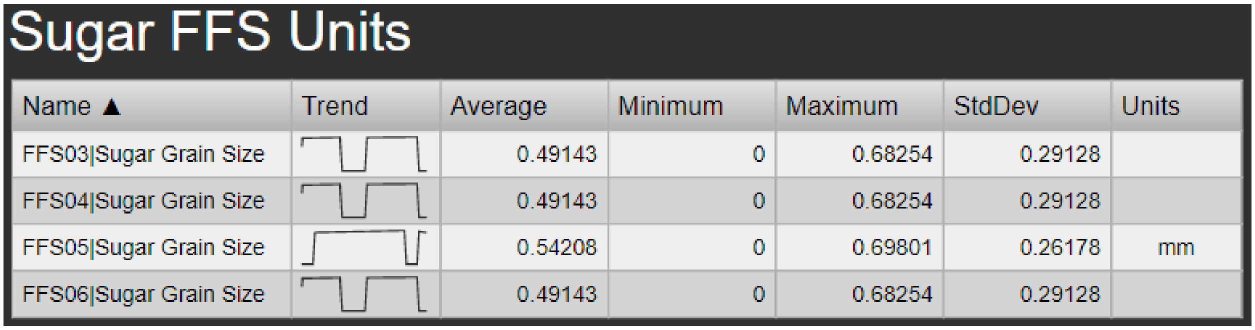
Step 7. Look at the table on the bottom-right corner, which gives us the Sugar Grain Size statistics for the sugar packaging units in the sandbox environment (FFS03 to FFS06).

Step 8. In order to better inspect the Trend column, set the time span to 1 hour.



Do you notice anything strange? We've found the 2nd issue to solve!

Issue # 2. Trends and statistics for FFS units 3,4 and 6 are identical.

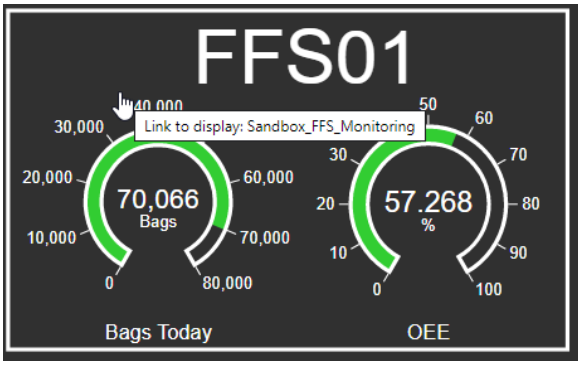


Sugar FFS Units

Name ▲	Trend	Average	Minimum	Maximum	StdDev	Units
FFS03|Sugar Grain Size		0.49143	0	0.68254	0.29128	
FFS04|Sugar Grain Size		0.49143	0	0.68254	0.29128	
FFS05|Sugar Grain Size		0.54208	0	0.69801	0.26178	mm
FFS06|Sugar Grain Size		0.49143	0	0.68254	0.29128	

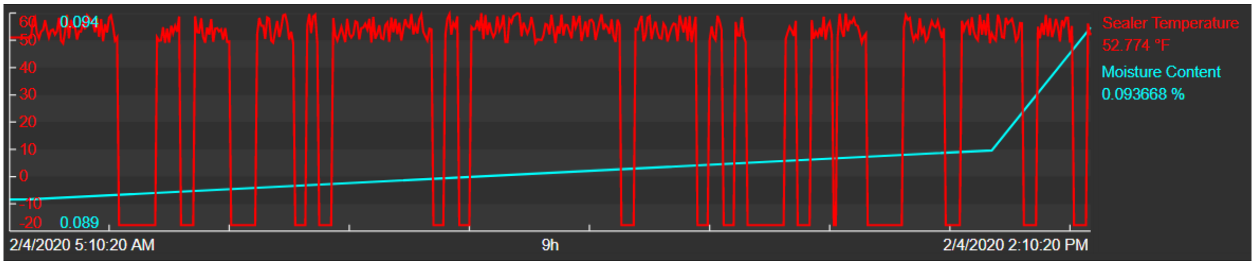
Let’s proceed in our diagnostics.

Step 9. Click on the tile for FFS01 to go back to the **Sandbox_FFS_Monitoring** display.

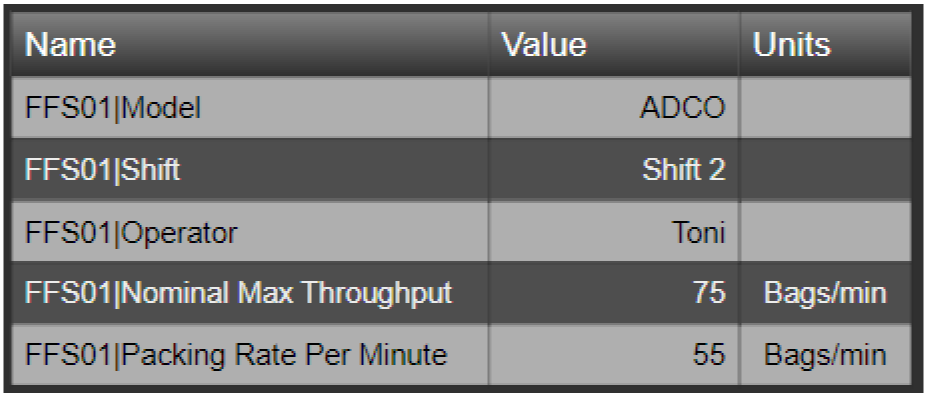


Look at the trend of the **Sealer temperature**. This temperature is expected to vary **between 70 and 270 °F**. Is it operating within the expected range? Auch, one more issue!

Issue # 3. The attribute Sealer temperature shows values outside the expected range.



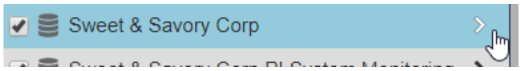
The chief maintenance engineer would like to include the **Last Service Date** attribute in the Table symbol.

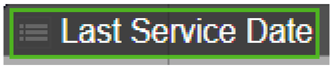


Name	Value	Units
FFS01|Model	ADCO	
FFS01|Shift	Shift 2	
FFS01|Operator	Toni	
FFS01|Nominal Max Throughput	75	Bags/min
FFS01|Packing Rate Per Minute	55	Bags/min

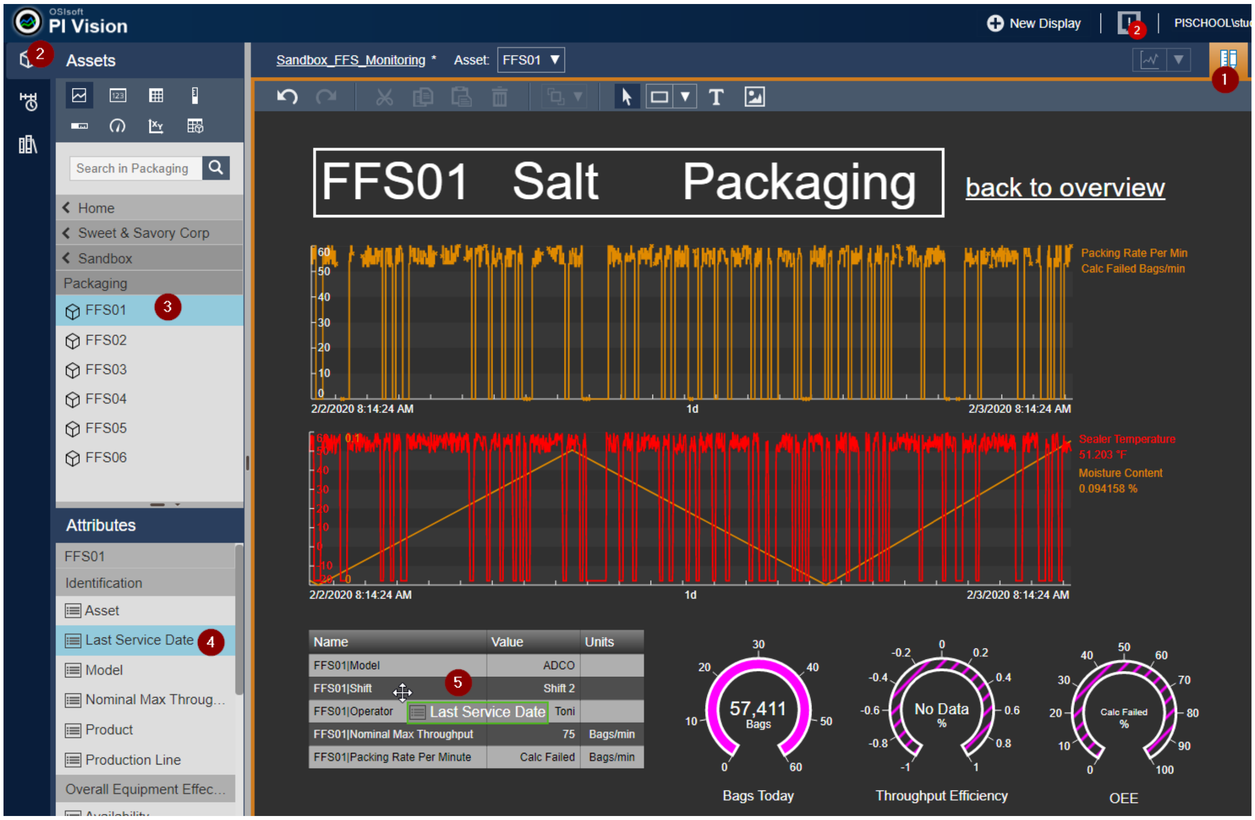
In order to add the Last Service Date attribute to the table (see also Figure 5):

1) Switch to Edit mode (button in the top-right corner of the display).
2) Open the asset pane on the left side of the display.
3) Navigate to **Sweet & Savory Corp → Sandbox → Packaging → FFS01**.

   (use the little arrows to drill down to the next level )

4) Select the **Last Service Date** attribute from the Attributes pane (bottom left pane)
5) Drag and drop it on top of the Table symbol (you should see it displayed as )

   NB: Do this operation only once! If the display doesn't update, please give it some more time.



Name	Value	Units
FFS01|Model	ADCO	
FFS01|Shift	Shift 2	
FFS01|Operator	Toni	
FFS01|Nominal Max Throughput	75	Bags/min
FFS01|Packing Rate Per Minute	Calc Failed	Bags/min

Figure 5. How to add an attribute to an existing symbol in PI Vision

Did you experience any delays when adding the attribute to the display? This brings us to the 3rd issue:

Issue # 4. The attribute Last Service Date slows down the loading of the PI Vision display.

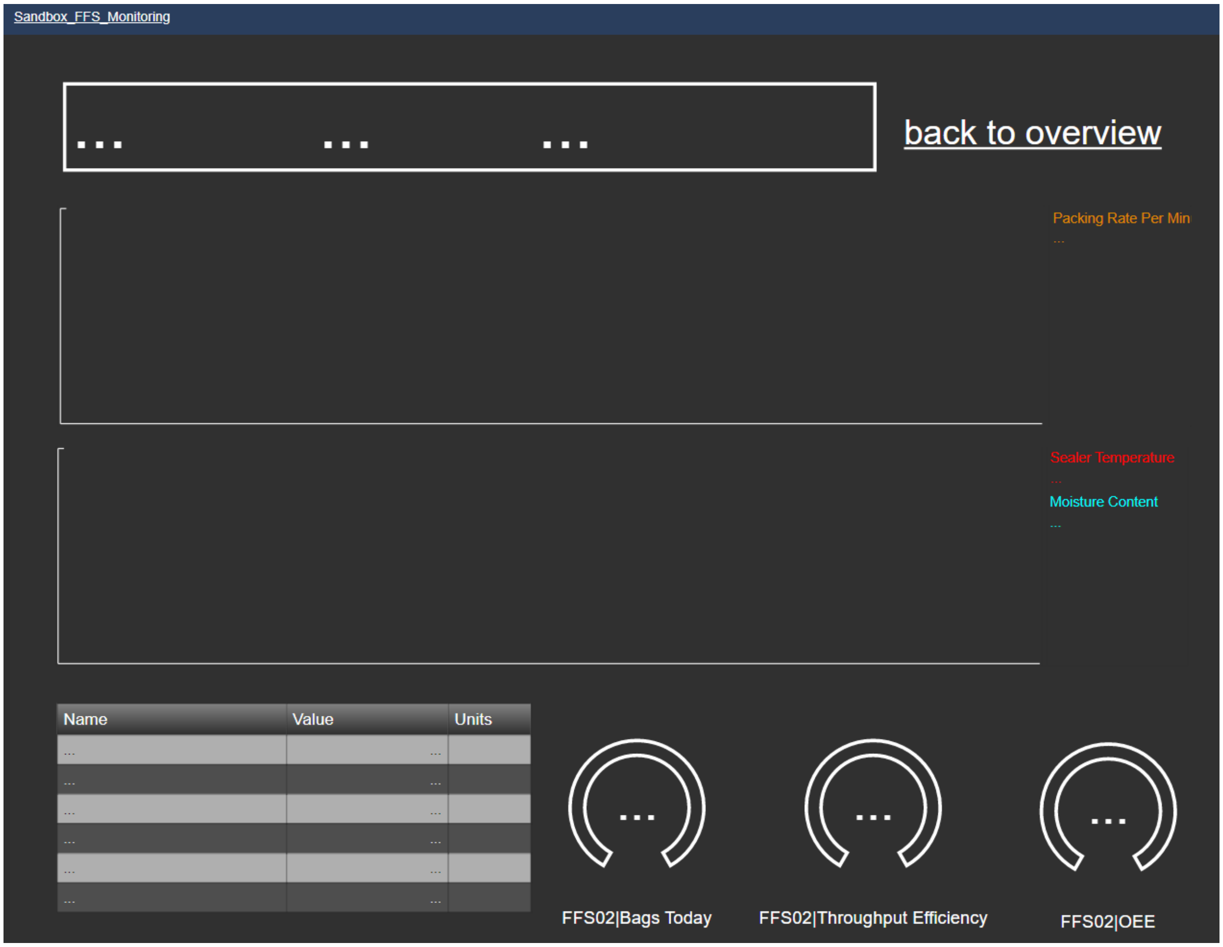
Some gory details below…

NERD ALERT – Data Caching in PI Vision

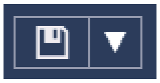
The performance issue above normally affects the initial loading of the display; the delay could seem insignificant when refreshing the page. This is due to data caching, a mechanism that helps reducing network workload by keeping a copy of frequently used data on the client machine.

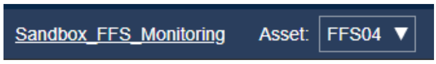
The loading time of a PI Vision display depends both on the caching of data on the PI Vision server itself, and on the caching of data on the web browser of the client machine. You can check the "fresh" loading time of our PI Vision display by clearing the web browser cache (*Ctrl + Shift + Del* → Clear Data) and the

PI Vision cache (e.g. by recycling the Application Pools or running an *iisreset* from an elevated command prompt). You should see that the Sandbox_FFS_Monitoring display takes a long time to fully load.



Name	Value	Units
...	...	
...	...	
...	...	
...	...	
...	...	
...	...	

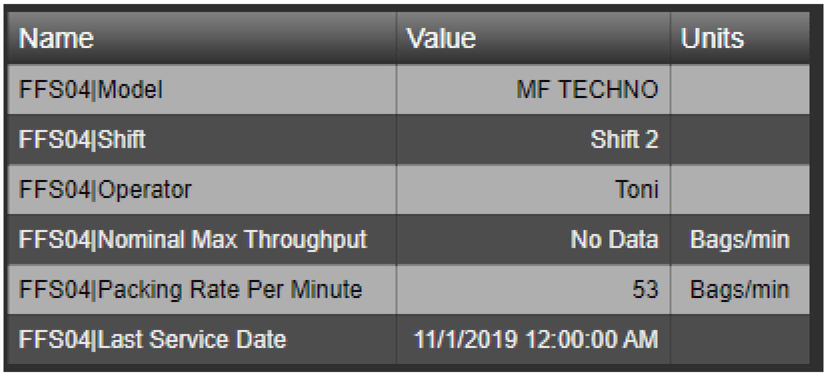
Step 10. Save the change to the display by clicking on the  button on the top-right corner of the display.

Step 11. Switch to unit FFS04 (using the Asset Dropdown Menu 

) and look at the table in the bottom-left corner.

Do you notice any problems?

Issue # 5. Asset FFS04 shows “No Data” for the Nominal Max Throughput.

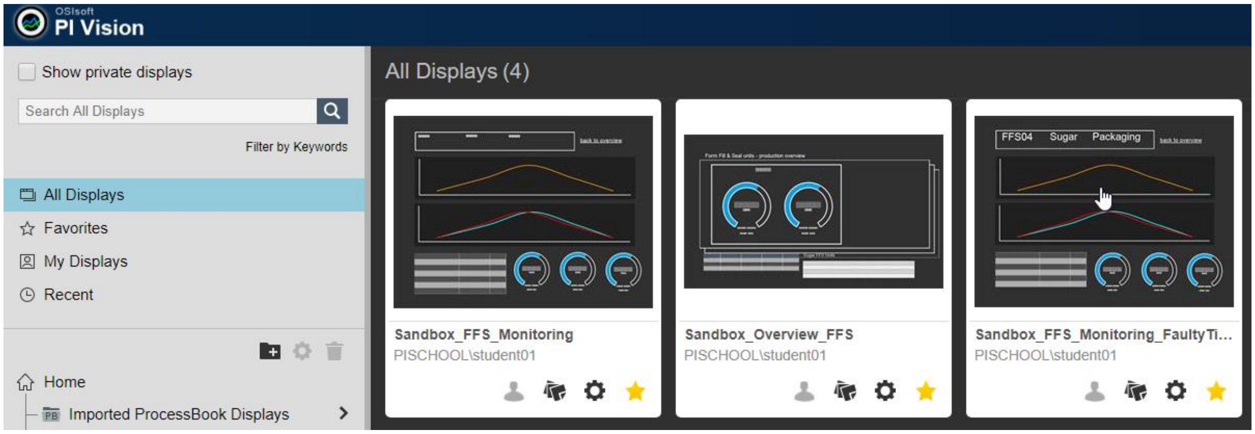


Name	Value	Units
FFS04|Model	MF TECHNO	
FFS04|Shift	Shift 2	
FFS04|Operator	Toni	
FFS04|Nominal Max Throughput	No Data	Bags/min
FFS04|Packing Rate Per Minute	53	Bags/min
FFS04|Last Service Date	11/1/2019 12:00:00 AM	

Let’s move on to the next issue and don’t worry… we will tackle these problems one by one in the coming sections!

Recently the chief maintenance engineer tried to build a new display, but he encountered difficulties with the display titles.

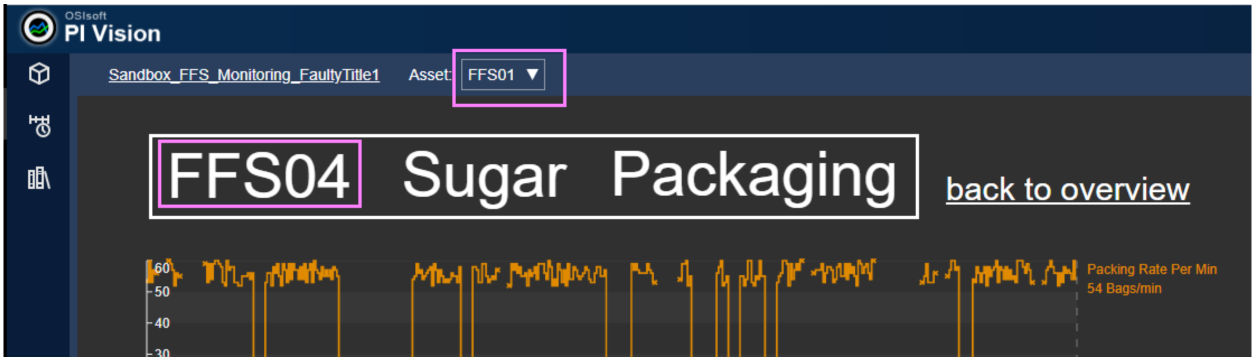
Step 12. Click on OSIsoft PI Vision and open the PI Vision display **Sandbox_FFS_Monitoring_Faulty_Title** (the displays are organized from left to right in order of last access, so this display could be on the rightmost of the main page if you haven’t opened it already).



Step 13. Switch between FFS units using the Asset Dropdown Menu.

What happens to the title of the display?

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset.



Now that we have a few issues to solve, let’s move to Asset Framework and sort them all out!

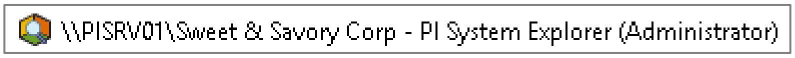
2.4. Asset Framework Best Practices - Directed activities

2.4.1. AF templates

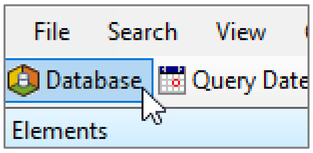
The first issue that we want to address is:

Issue # 1. Asset FFS05 is not available in the PI Vision Asset Dropdown Menu

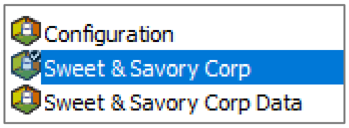
Step 1. Open **PI System Explorer** from the taskbar, make sure that you are in the database **Sweet & Savory Corp** .


\\PISRV01\Sweet & Savory Corp - PI System Explorer (Administrator)

1.b. If you are not connected to this database, move to **Database**

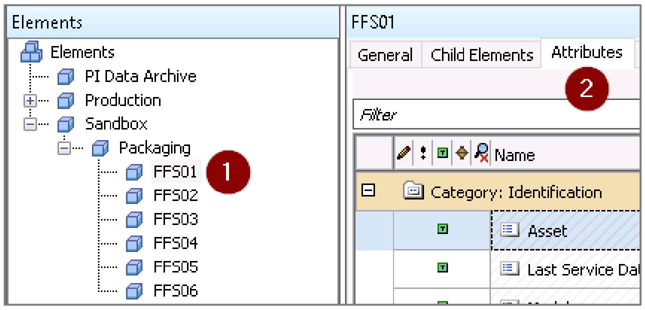


1.c. Double-click on the **Sweet & Savory Corp** database .

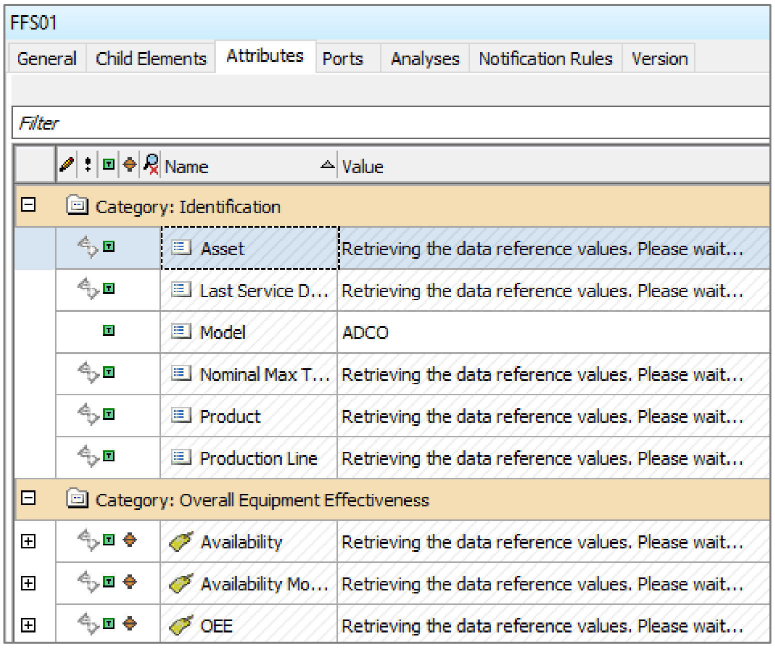


Step 2. Expand the **Sandbox** branch by clicking on the ⊞.

Step 3. Select **FFS01** and open the Attributes tab.



It might take some time for the Attributes tab to get populated with values (Figure 6). We will come back to this issue later on in the Lab (Section 2.4.3).

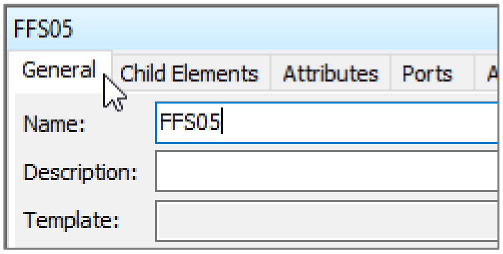


Name	Value
Category: Identification	
Asset	Retrieving the data reference values. Please wait…
Last Service D…	Retrieving the data reference values. Please wait…
Model	ADCO
Nominal Max T…	Retrieving the data reference values. Please wait…
Product	Retrieving the data reference values. Please wait…
Production Line	Retrieving the data reference values. Please wait…
Category: Overall Equipment Effectiveness	
Availability	Retrieving the data reference values. Please wait…
Availability Mo…	Retrieving the data reference values. Please wait…
OEE	Retrieving the data reference values. Please wait…

Figure 6. PSE hangs with message "Retrieving the data reference values. Please wait…"

Step 4. Switch to FFS05 to compare the attribute list. What do you notice? (see also Table 2)

Step 5. Click on the General tab and compare the two elements. (see also Table 2)

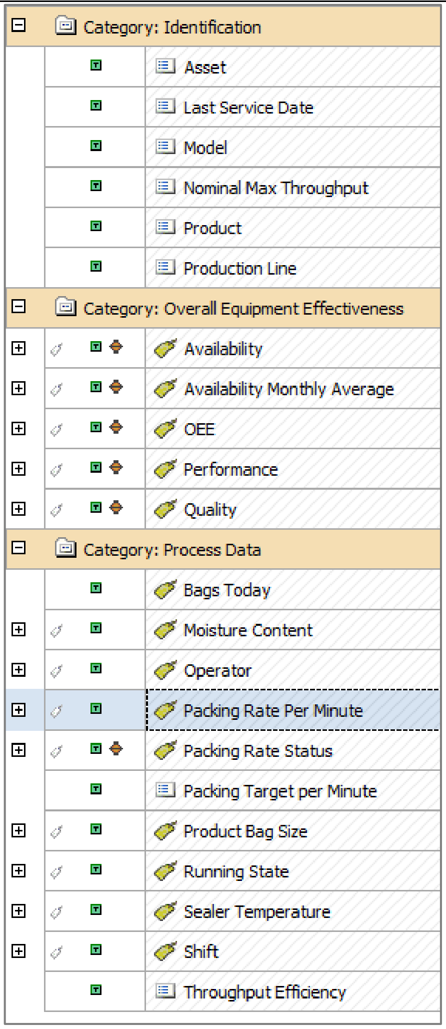
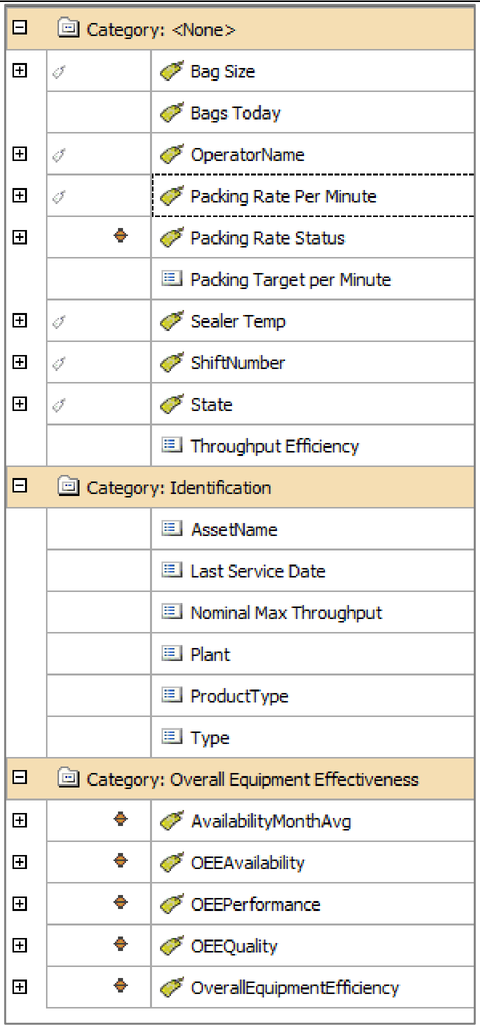
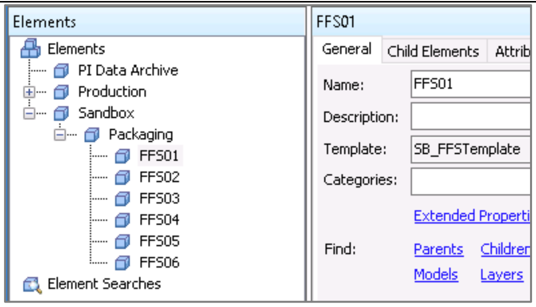
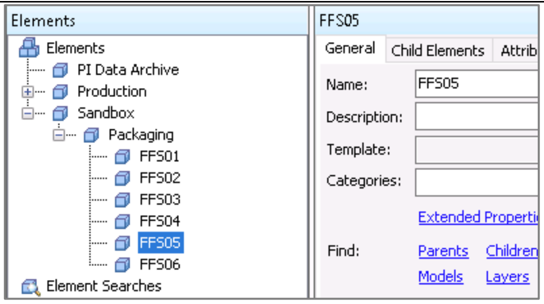


Name: FFS05

Description:

Template:

Table 2. Comparison between FFS01 and FFS05

FFS01	FFS05
Attributes tab	**Attributes tab**
 Category: Identification Asset Last Service Date Model Nominal Max Throughput Product Production Line Category: Overall Equipment Effectiveness Availability Availability Monthly Average OEE Performance Quality Category: Process Data Bags Today Moisture Content Operator Packing Rate Per Minute Packing Rate Status Packing Target per Minute Product Bag Size Running State Sealer Temperature Shift Throughput Efficiency	 Category: <None> Bag Size Bags Today OperatorName Packing Rate Per Minute Packing Rate Status Packing Target per Minute Sealer Temp ShiftNumber State Throughput Efficiency Category: Identification AssetName Last Service Date Nominal Max Throughput Plant ProductType Type Category: Overall Equipment Effectiveness AvailabilityMonthAvg OEEAvailability OEEPerformance OEEQuality OverallEquipmentEfficiency
General tab	**General tab**
 Elements Elements PI Data Archive Production Sandbox Packaging FFS01 FFS02 FFS03 FFS04 FFS05 FFS06 Element Searches FFS01 General | Child Elements | Attrib Name: FFS01 Description: Template: SB_FFSTemplate Categories: Extended Properti Find: Parents Childrer Models Layers	 Elements Elements PI Data Archive Production Sandbox Packaging FFS01 FFS02 FFS03 FFS04 FFS05 FFS06 Element Searches FFS05 General | Child Elements | Attrib Name: FFS05 Description: Template: Categories: Extended Properti Find: Parents Children Models Layers

You probably observed that the two elements do not show exactly the same attributes and do not use the same attribute naming convention. In FFS01 you'll notice a small T symbol next to each attribute, which means that the attribute is based on an AF element template. FFS01 is based on the AF element template **Sandbox_FFS** (visible under the General tab), while FFS05 does not use any template.

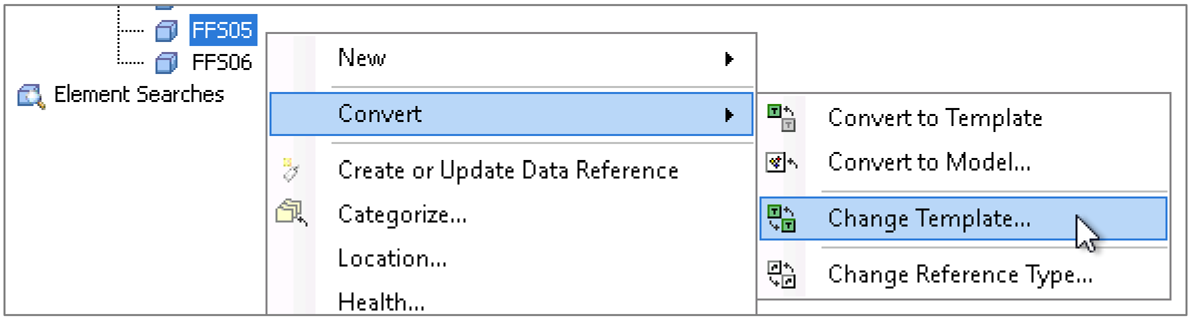
This discrepancy is due to a workaround adopted in the modelling phase: FFS05 has some issues with the PLC, and not all readings are available. In particular, the moisture content is not available. For this reason, the data management team decided to build FFS05 from scratch (without using the AF element template) and created only the attributes for which measurements are available.

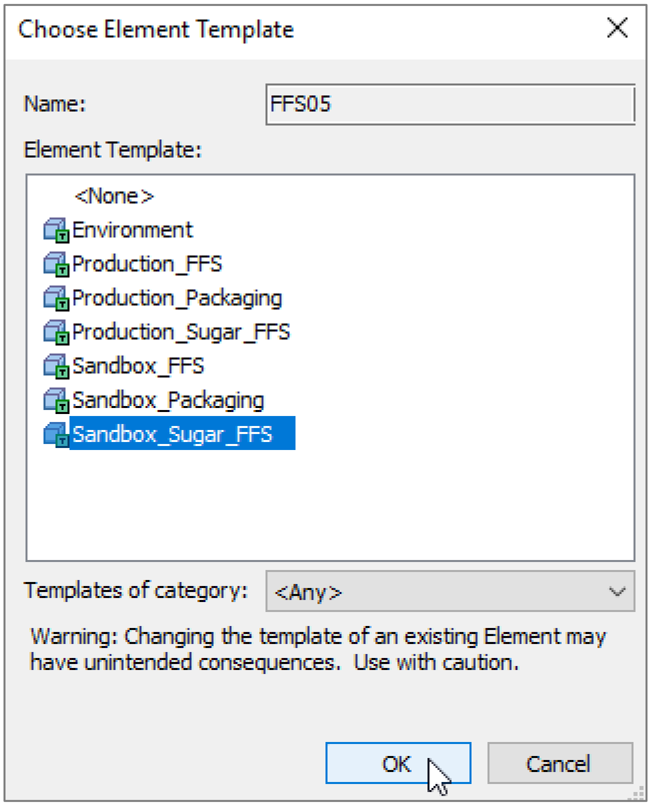
The problem is that, over a few months' time, the structure and nomenclature of the AF element template was modified compared to the manually created element FFS05.

Creating AF elements without using an AF element template presents several drawbacks. For example, FFS05 will not be recognized by PI Vision as an FFS unit, and any change made to the FFS AF element template will not propagate to FFS05.

A better solution is to (a) assign FFS05 to the same AF element template as its siblings and (b) exclude the attributes that are not available for FFS05.

Step 6. Right click on FFS05 and select Convert → Change Template → **Sandbox_Sugar_FFS** → OK




Choose Element Template

Name: FFS05

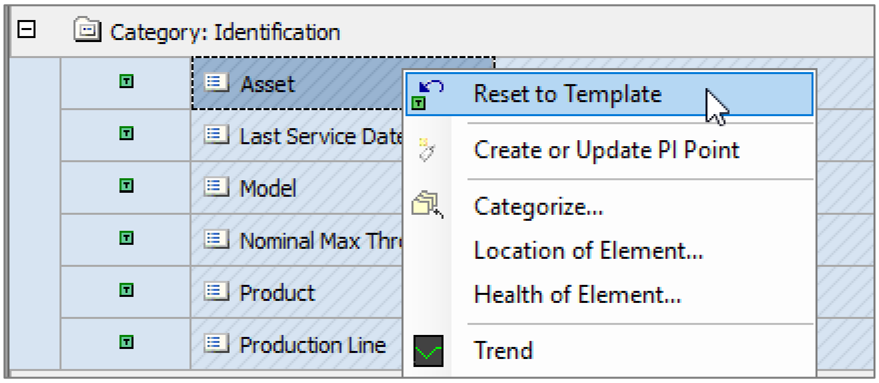
Element Template:

<None>
Environment
Production_FFS
Production_Packaging
Production_Sugar_FFS
Sandbox_FFS
Sandbox_Packaging
Sandbox_Sugar_FFS

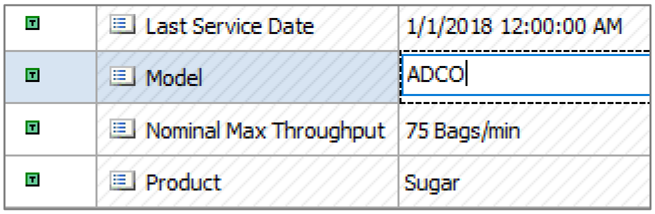
Templates of category: <Any>

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

Step 7. Select all the attributes of element FFS05 (you can select the first one, then keep Shift down and select the last one), right-click and choose "**Reset to Template**"



Step 8. Type in the name of the Model for FFS05 (we will come back to this in Section 2.4.5)

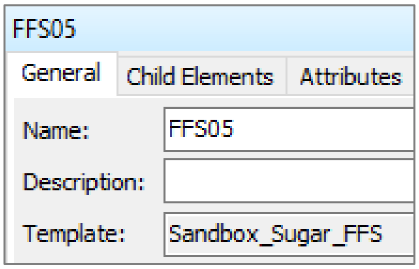


🟩	Last Service Date	1/1/2018 12:00:00 AM
🟩	Model	ADCO
🟩	Nominal Max Throughput	75 Bags/min
🟩	Product	Sugar

Excluded attribute property

Notice that the attribute list now matches what you see in the template configuration.

Also, when clicking on the General tab you'll see the AF element template used:



Name: FFS05

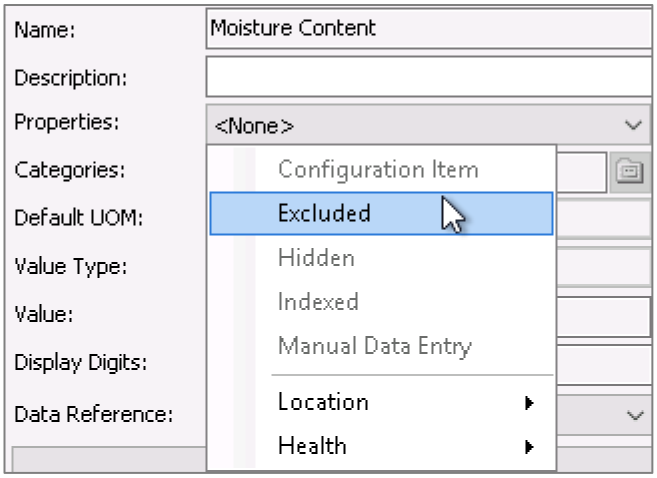
Description:

Template: Sandbox_Sugar_FFS

In order to exclude the attribute Moisture Content, not available for FFS05:

Step 9. Select the attribute **Moisture Content**, move to the right pane and select **Excluded** from the Properties dropdown menu.

More information on the **Excluded** attribute property is available in the LiveLibrary.



Name: Moisture Content

Description:

Properties: <None>

Categories:

Default UOM:

Value Type:

Value:

Display Digits:

Data Reference:

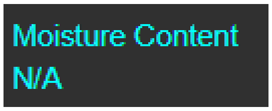
Step 10. Important: Click on Check In **to save your changes.**

Step 11. Move back to **PI Vision**, open the display **Sandbox_FFS_Monitoring**, refresh the page and verify that FFS05 is visible under the Asset Dropdown Menu.

NB: It might take a few minutes for your PI Vision server to get updated with the latest changes on the AF Server (technically, PI Vision relies on the PI Web API Crawler to crawl the AF databases, and the crawling happens by default every 3 minutes. We lowered the crawling refresh interval to 1 minute for the purpose of this Lab).

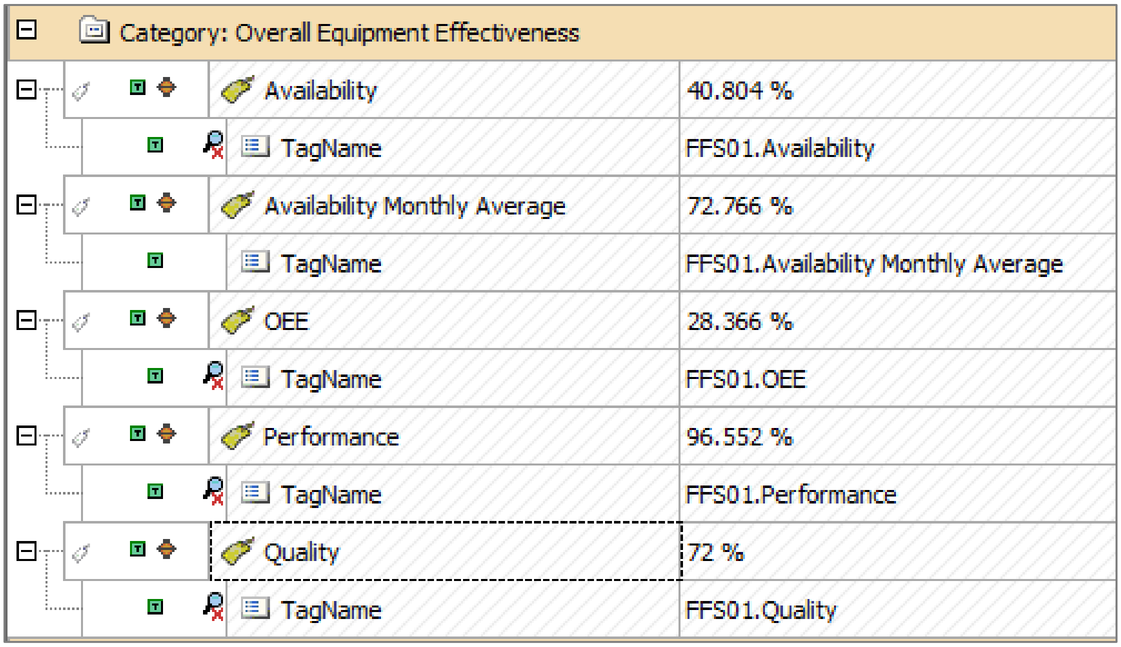
Step 12. Verify what happens to the Moisture Content for element FFS05.

[it should show *N/A*, not available]



Moisture Content
N/A

You might have noticed that each PI Point data reference attribute (i.e. each AF attribute pointing to a PI Point on the Data Archive, indicated by a symbol) has a child attribute called *TagName*. Also, these attributes have a symbol next to them:



Category: Overall Equipment Effectiveness

Availability	40.804 %
TagName	FFS01.Availability
Availability Monthly Average	72.766 %
TagName	FFS01.Availability Monthly Average
OEE	28.366 %
TagName	FFS01.OEE
Performance	96.552 %
TagName	FFS01.Performance
Quality	72 %
TagName	FFS01.Quality

Would you be able to tell (a) why these attributes show a symbol and (b) why these attributes exist in the first place? This topic is a bit hairy, so we'll probably skip it during the Lab. Feel free to go through it at your own pace.

NERD ALERT - Hidden attribute property

The symbol is used to indicate that these attributes are hidden.

A hidden attribute is one that cannot be retrieved by client applications such as PI Vision. This property is useful if an attribute is used to hold an **intermediate result** or a configuration parameter. For example, you could hide string values that are concatenated to build a tag name.

It can also be useful to set an attribute property to Hidden in a template when configuration has not been fully completed in the element itself. For example, elements are being created from a template with a PI Point data reference, but because some instrumentation is missing, PI Points do not yet exist for all the elements. By setting the attributes for the missing instrumentation to Hidden, a client application is prevented from obtaining an error result from a search (adapted from LiveLibrary).

In our case, the TagName attributes are hidden simply because they don't need to be available in PI Vision, as they are used for AF configuration purposes.

NERD ALERT - Referencing TagName child attributes

The TagName attributes have been used to store the name of the PI Point associated with the parent AF attributes (e.g. Availability, OEE, …).

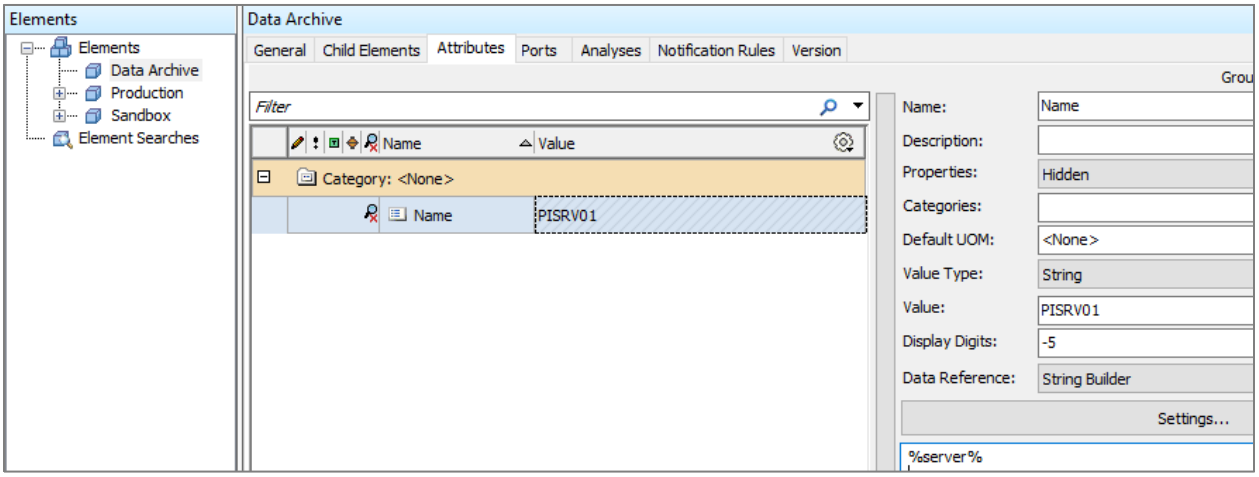
The reason is the following: suppose that many of your PI Points don't follow a clear naming convention. As an example, the Sealer Temperature of FFS01 could be associated with PI Point *FFS01SealTemp*, while the same attribute under FFS02 could be associated with PI Point *02ST3000*. Although this could seem illogical, it often occurs whenever the PI Points are named after the original data source items, which in turn follow the naming convention of the various PLC/SCADA manufacturers.

If you find yourself unable to establish a simple pattern that automates the associations between AF attributes and PI Points (e.g. using substitution parameters), you need to manually associate the AF attributes with the corresponding PI Points. The TagName child attributes allow you to "push" the PI Point names (e.g. *FFS01.OEE*) into the configuration of your PI Point data reference attributes (e.g. Sealer Temperature, Moisture Content etc.). To accelerate this operation, you could use **PI Builder**, an Excel Add-In that allows you to edit attribute configurations in bulk. For more information on using PI Builder, watch the playlist available on the OSIsoft Learning channel. In order to facilitate the system's maintenance, you can configure the TagName attribute as a Table Lookup, and store all PI Point names in an AF Table (more on AF Tables in Section 2.4.4).

NERD ALERT - Data Archive element

You might have noticed an AF element called *Data Archive* (we'll use italic font style to distinguish the element name from the actual Data Archive software component).

What is this element doing here, and why does it show an attribute containing the name of our Data Archive?



Name	Value
Name	PISRV01

Name: Name
Description:
Properties: Hidden
Categories:
Default UOM: <None>
Value Type: String
Value: PISRV01
Display Digits: -5
Data Reference: String Builder
%server%

The AF element *Data Archive* is simply a placeholder. It allows you to reference all the PI Point data reference attributes (e.g. Sealer Temperature, Moisture Content etc.) to the Data Archive where the PI Points reside. More importantly, if you need to **export the AF database and import it into another PI System**, you'll only need to modify the value of the Name attribute under the *Data Archive* element, and all the PI Point data reference attributes will point to the new Data Archive.

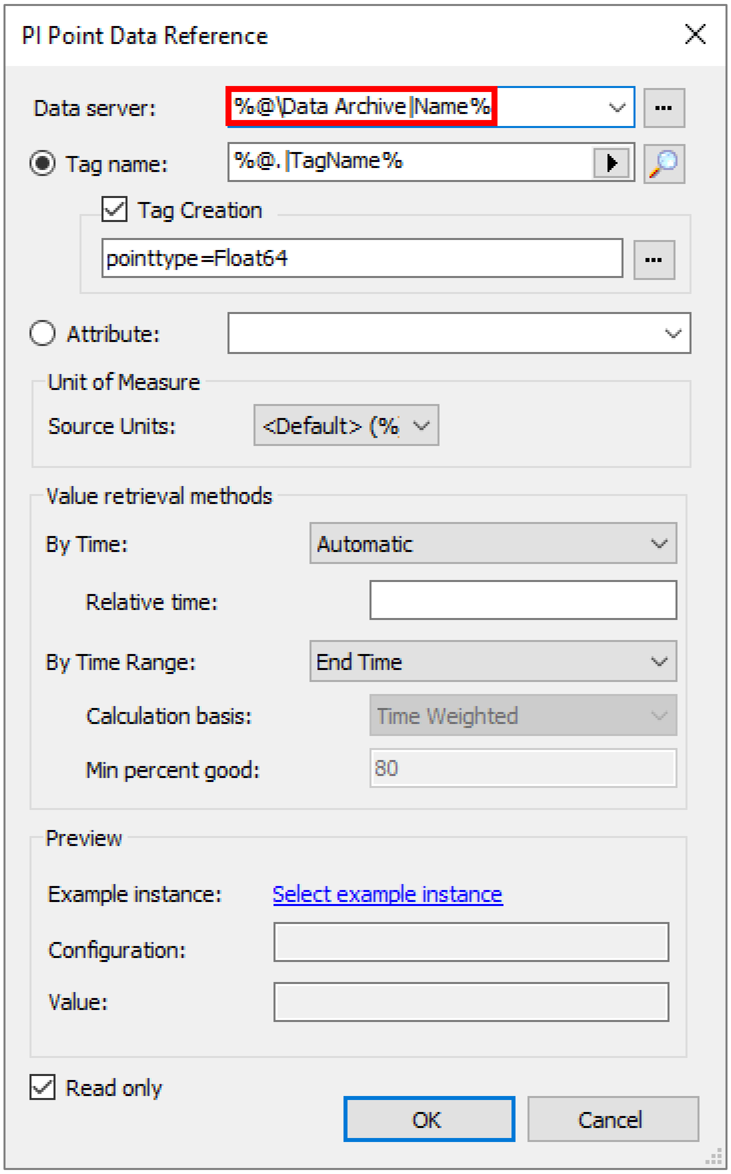
A typical use case is when you need to move a database from a production server to a sandbox server or vice versa. Most likely the Data Archive in the sandbox environment is named differently with respect to the Data Archive in the Production environment. Editing the value of the Name attribute under the Data Archive can save you a lot of time compared to editing hundreds or thousands of attribute configurations.

How is this behavior obtained? Take for example the Sandbox_FFS template and navigate to the Availability attribute. Click on Settings and inspect the "Data server" field (see red box in Figure 7). You'll notice that it points to the Name attribute under the *Data Archive* element using syntax:

%@\Data Archive|Name%

You'll notice also that the Tag Name field is pointing to the child attribute TagName, which we discussed before, using syntax:

%@.|TagName%


PI Point Data Reference

Data server: %@\Data Archive|Name%

● Tag name: %@.|TagName%

☑ Tag Creation

pointtype=Float64

○ Attribute:

Unit of Measure

Source Units: <Default> (%

Value retrieval methods

By Time: Automatic

Relative time:

By Time Range: End Time

Calculation basis: Time Weighted

Min percent good: 80

Preview

Example instance: Select example instance

Configuration:

Value:

☑ Read only

Figure 7. PI Point data reference configuration using substitution parameters

For more information on different ways of pointing AF attributes to PI Points, please watch:

[OSIsoft: PI Point data references Beginner to Advanced [v2.9.5.8368]]

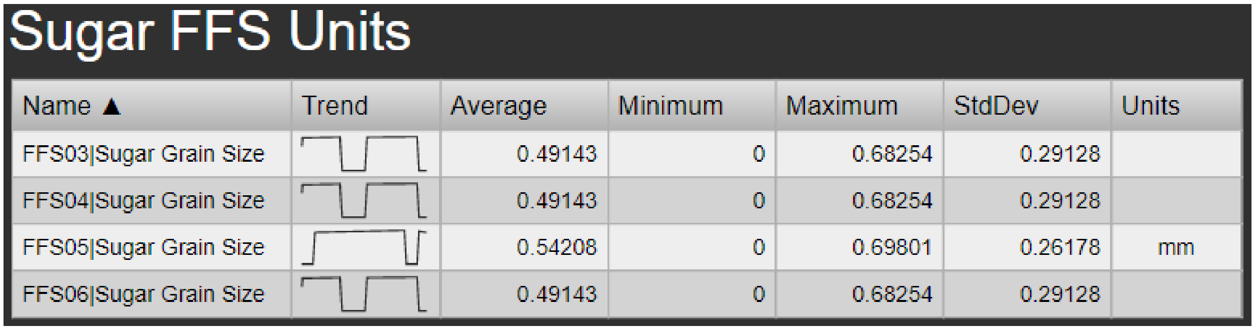
So far, we saw that one way of handling variations among assets is using the Excluded attribute property. This is useful in situations where not all attributes in an AF element template apply (in our example, the Moisture Content was not available for FFS05).

What happens when the **same variation** applies to **several elements** from the same element template? For example, if 5 out of 100 elements need an extra attribute, do I need to add the extra attribute to the element template used by all 100 elements and then exclude it from 95 of them? We will now see how derived templates help us handle this type of variations.

2.4.2. More on handling variations – Allow Extensions vs Derived Templates

Remember the second issue observed in PI Vision?

Issue # 2. Trends and statistics for FFS units 3,4 and 6 are identical



Sugar FFS Units

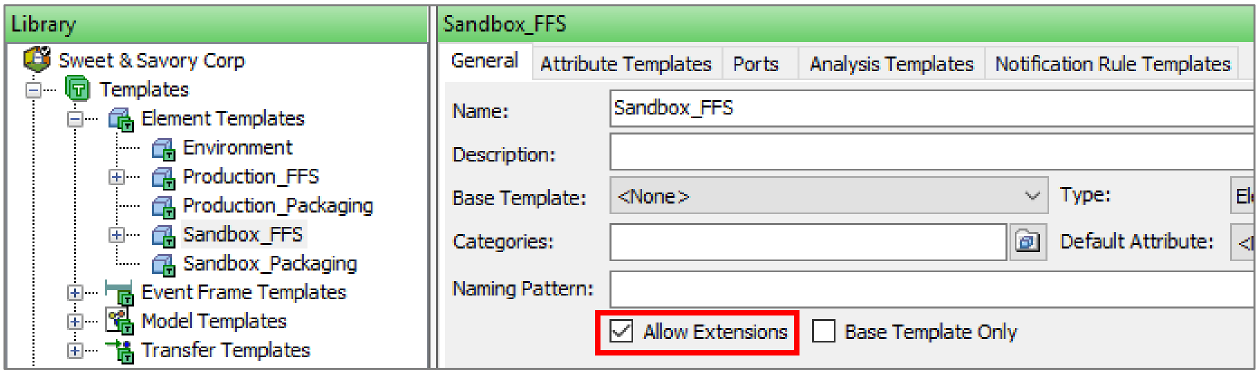
Name ▲	Trend	Average	Minimum	Maximum	StdDev	Units
FFS03|Sugar Grain Size		0.49143	0	0.68254	0.29128	
FFS04|Sugar Grain Size		0.49143	0	0.68254	0.29128	
FFS05|Sugar Grain Size		0.54208	0	0.69801	0.26178	mm
FFS06|Sugar Grain Size		0.49143	0	0.68254	0.29128	

Let’s retrace how these AF attributes were configured in Sweet & Savory Corp’s AF structure.

In order to comply with the company’s quality standards, the **sugar** packed by the FFS units must meet strict requirements in terms of **grain size**. In out example, this is **not required for salt** packaging.

Allow Extensions

When creating the attribute Sugar Grain Size, the data management team enabled the option “**Allow Extensions**” (available under the General tab of the element template **Sandbox_FFS**):



Name: Sandbox_FFS

Description:

Base Template: <None>

Type: El

Categories:

Default Attribute: <

Naming Pattern:

☑ Allow Extensions ☐ Base Template Only

This option allows you to add attributes to one or more individual AF elements without propagating the changes to all AF elements based on the same template.

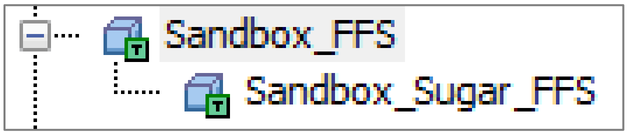
Despite its **flexibility** and its usefulness **in testing phases**, an incorrect use of the Allow Extension option often leads to a **lack of consistency** among elements, and results in manual operations whenever the extra attributes need to be added to another element.

In our specific case, a mistake was made in the attribute configuration: the Sugar Grain Size attributes of units FFS03, FFS04 and FFS06 are all associated with the same PI Point on the Data Archive. In other words, all attributes Sugar Grain Size are looking at the same source data, the data for FFS03. This is a very common “copy-paste” typo!

Derived Templates

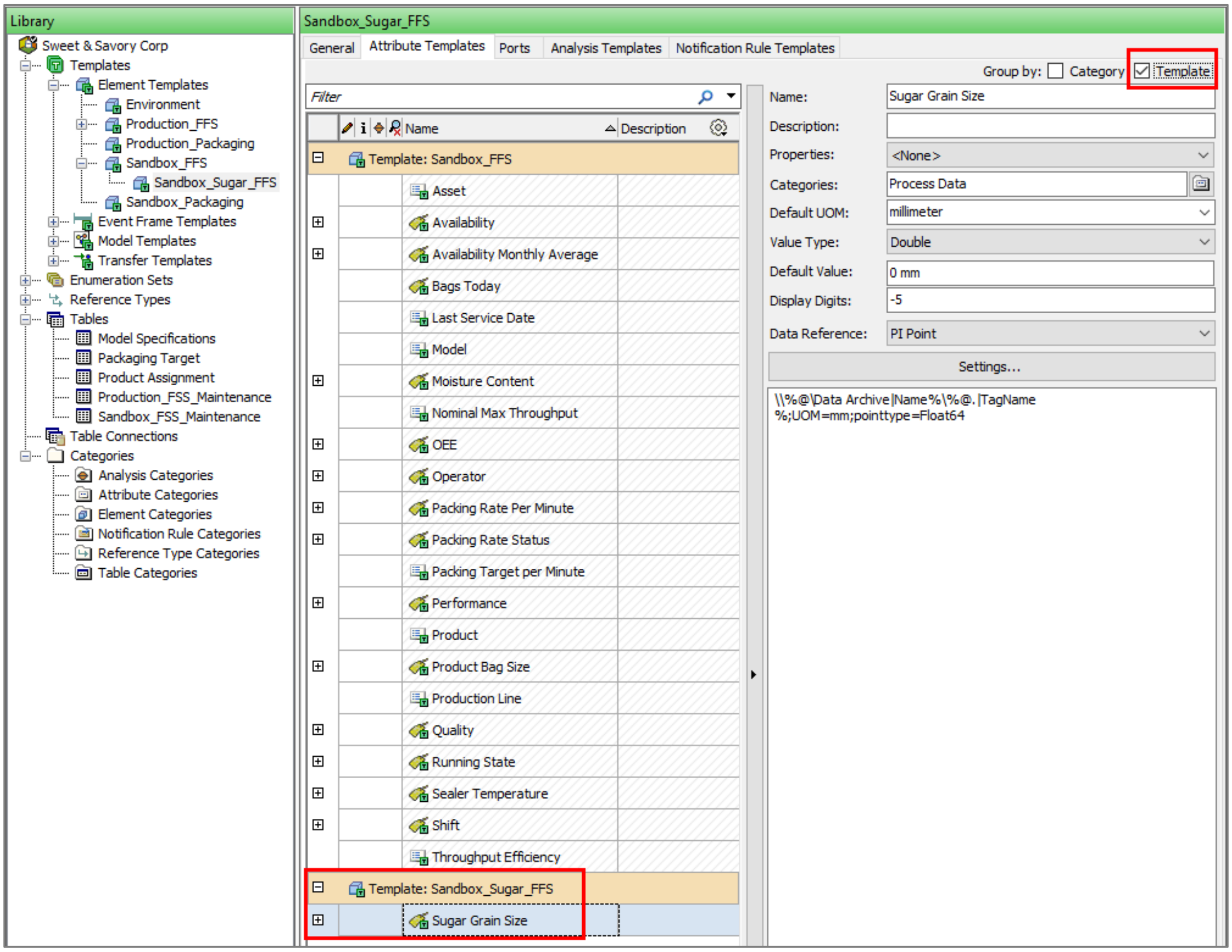
A better way to create additional attributes for one or more elements is to use **Derived Templates**.

Expand **Sandbox_FFS** (our base template), you will see a derived template called **Sandbox_Sugar_FFS.**



By default, the attribute templates tab of derived templates shows only the attributes that do not exist on the base template, or attributes for which we want to override the configuration compared to the base template.

Once you select a derived template, you can display also the attributes related to its Base Template by ticking **Group by: Template**:



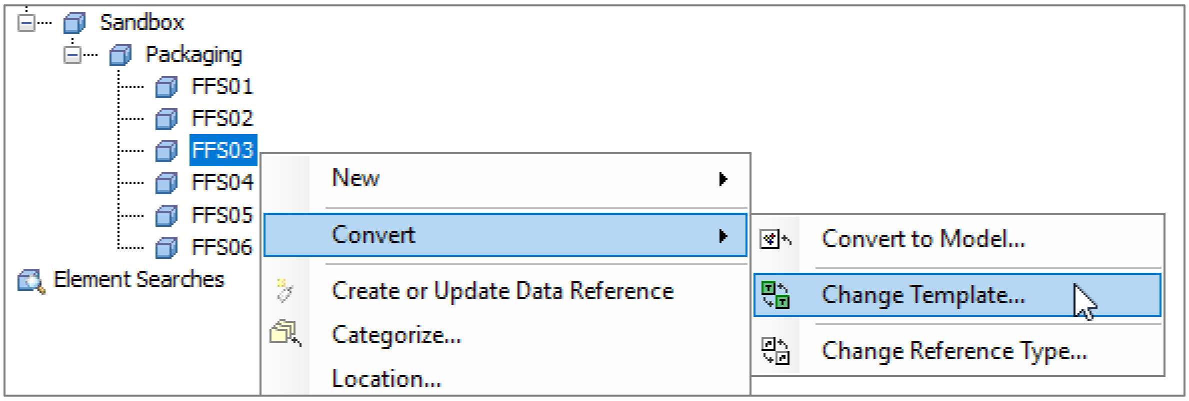
Name: Sugar Grain Size
Description:
Properties: <None>
Categories: Process Data
Default UOM: millimeter
Value Type: Double
Default Value: 0 mm
Display Digits: -5
Data Reference: PI Point

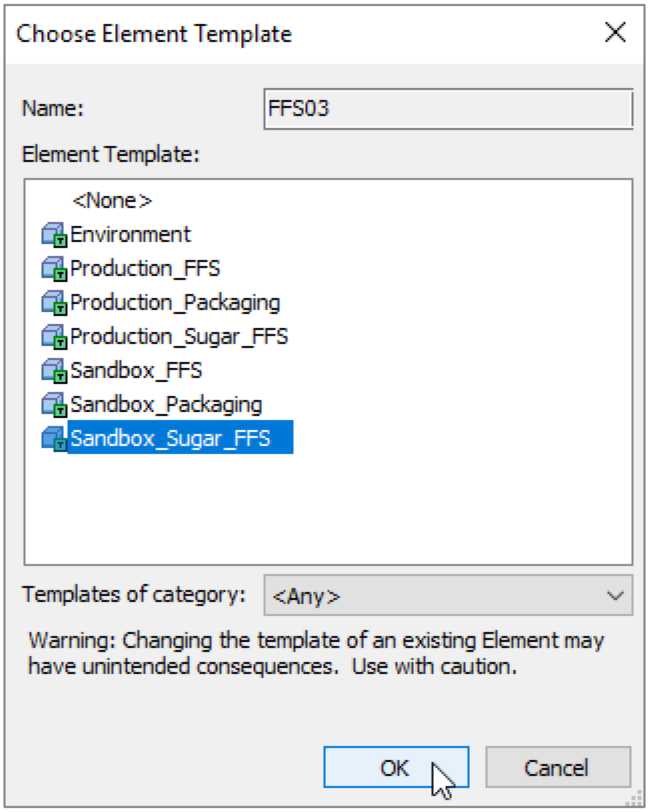
```
\\%@\Data Archive|Name%\%@.|TagName
%;UOM=mm;pointtype=Float64
```

In our case, the derived template includes an additional attribute called **Sugar Grain Size**. We can use this derived template instead of recurring to the *Allow Extensions* option.

Step 1. Go to the Elements tab, navigate to Sandbox → Packaging → **FFS03**

Step 2. Right click on FFS03 and select Convert → Change Template → **Sandbox_Sugar_FFS**





Choose Element Template

Name: FFS03

Element Template:

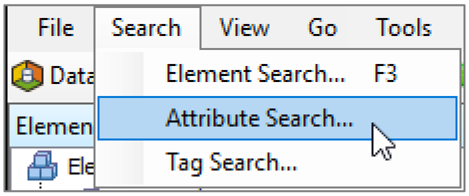
<None>
Environment
Production_FFS
Production_Packaging
Production_Sugar_FFS
Sandbox_FFS
Sandbox_Packaging
Sandbox_Sugar_FFS

Templates of category: <Any>

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

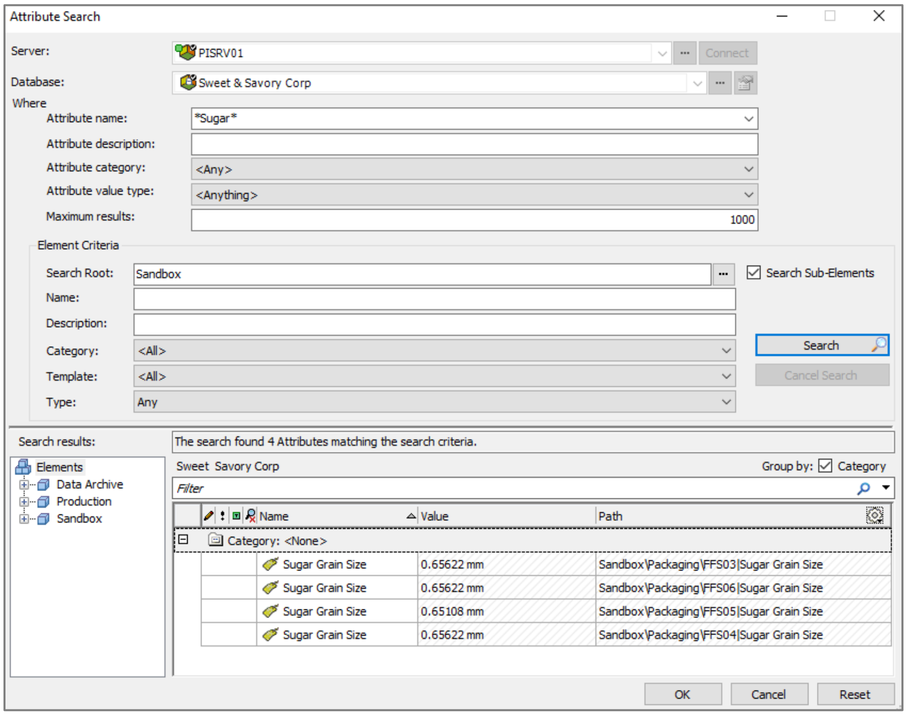
We now need to reset the Sugar Grain Size attributes to the template configuration. This is necessary to force AF to remap the attribute to the correct PI Point!

Step 3. Click on Search → Attribute Search…



Step 4. Use as filters:

- Attribute Name: **Sugar***
- Search Root: **Sandbox**



Server: PISRV01

Database: Sweet & Savory Corp

Where

Attribute name: *Sugar*

Attribute description:

Attribute category: <Any>

Attribute value type: <Anything>

Maximum results: 1000

Element Criteria

Search Root: Sandbox ☑ Search Sub-Elements

Name:

Description:

Category: <All>

Template: <All>

Type: Any

Search results: The search found 4 Attributes matching the search criteria.

Elements
- Data Archive
- Production
- Sandbox

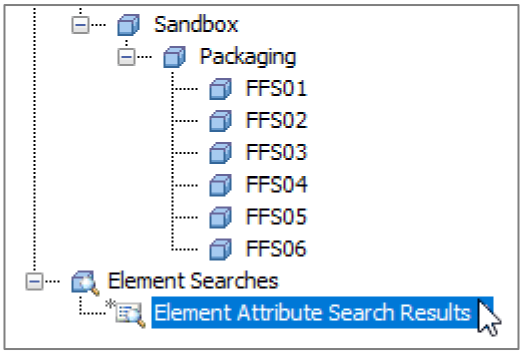
Sweet Savory Corp — Group by: ☑ Category

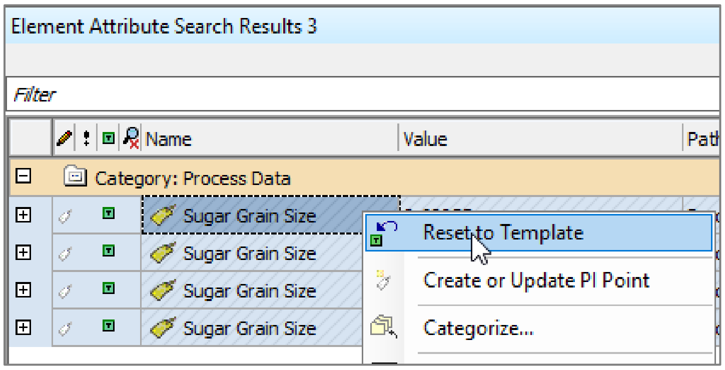
Filter

Category: <None>

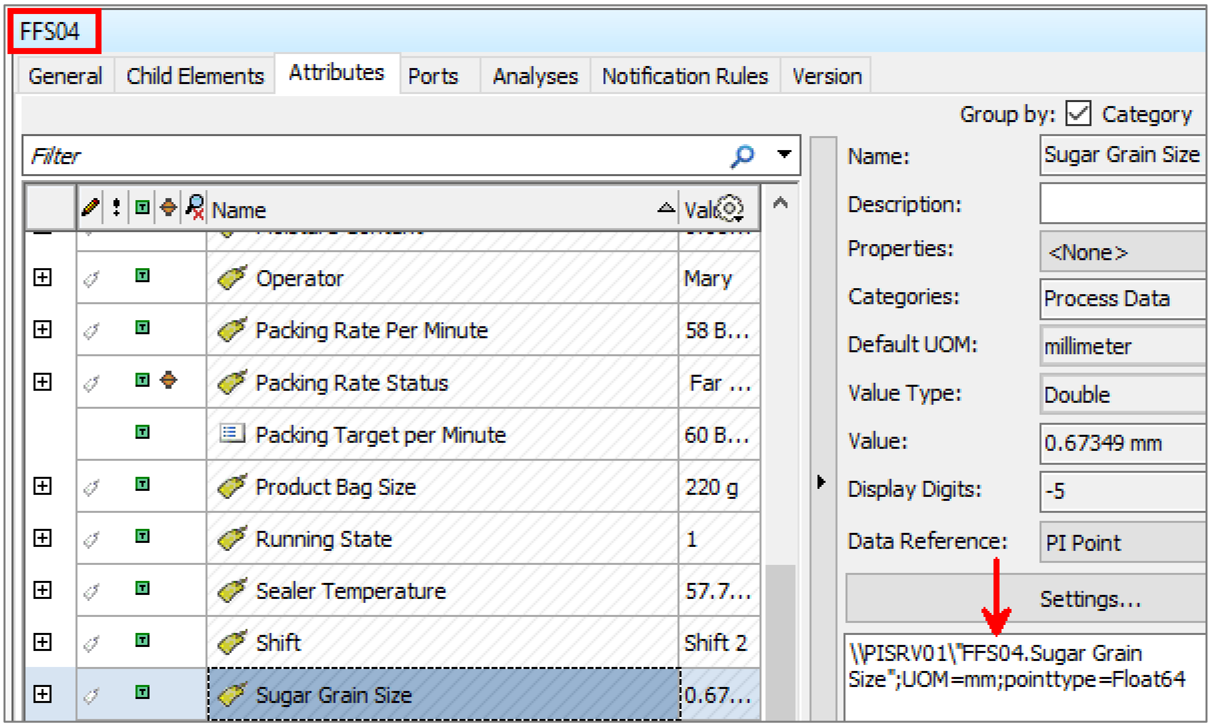
Name	Value	Path
Sugar Grain Size	0.65622 mm	Sandbox\Packaging\FFS03|Sugar Grain Size
Sugar Grain Size	0.65622 mm	Sandbox\Packaging\FFS06|Sugar Grain Size
Sugar Grain Size	0.65108 mm	Sandbox\Packaging\FFS05|Sugar Grain Size
Sugar Grain Size	0.65622 mm	Sandbox\Packaging\FFS04|Sugar Grain Size

Step 5. Select the **Element Attribute Search Results** (as in screenshot below) and select all the attributes; right-click and select **Reset to Template**.





You should see that the Sugar Grain Size now shows the correct PI Point (for example, FFS04 will have its Sugar Grain Size attribute pointing to the PI Point *FFS04.Sugar Grain Size*).



Name	Value
Operator	Mary
Packing Rate Per Minute	58 B…
Packing Rate Status	Far …
Packing Target per Minute	60 B…
Product Bag Size	220 g
Running State	1
Sealer Temperature	57.7…
Shift	Shift 2
Sugar Grain Size	0.67…

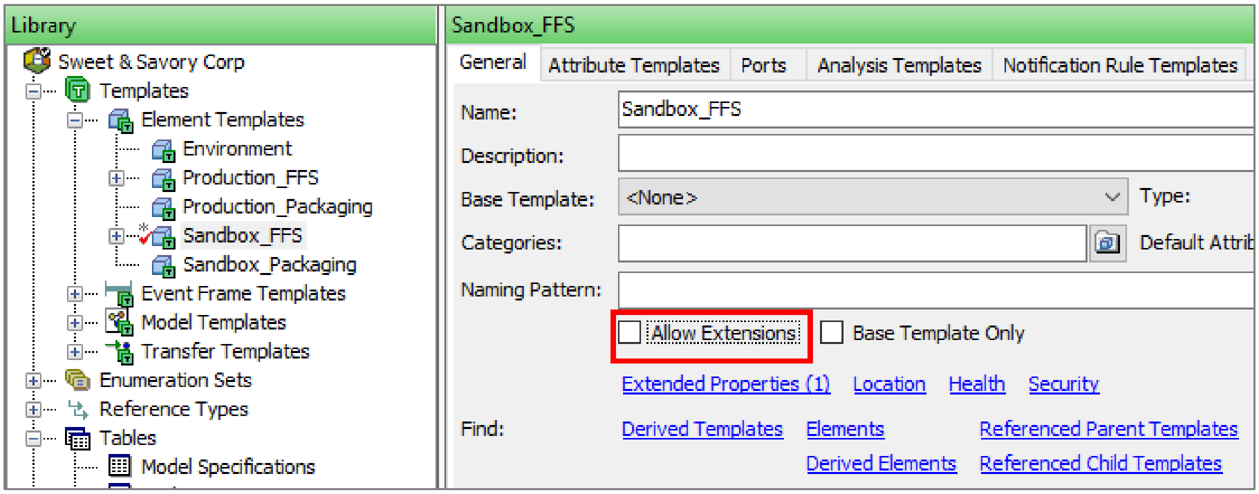
Name: Sugar Grain Size
Description:
Properties: <None>
Categories: Process Data
Default UOM: millimeter
Value Type: Double
Value: 0.67349 mm
Display Digits: -5
Data Reference: PI Point

\\PISRV01\"FFS04.Sugar Grain Size";UOM=mm;pointtype=Float64

Step 6. Move between elements and make sure that all attributes are populated with live data. Again, you'll notice a small 🟩 symbol next to each attribute, which means that the attribute is templatized.

Step 7. Move to the Library tab, select template **Sandbox_FFS** (Figure 8).

Step 8. Under the General tab, untick the "Allow Extensions" checkbox (this will prevent accidental/undesired creation of attributes at the level of a single element)



Library

- Sweet & Savory Corp
  - Templates
    - Element Templates
      - Environment
      - Production_FFS
      - Production_Packaging
      - Sandbox_FFS
      - Sandbox_Packaging
    - Event Frame Templates
    - Model Templates
    - Transfer Templates
  - Enumeration Sets
  - Reference Types
  - Tables
    - Model Specifications

Sandbox_FFS

Name: Sandbox_FFS

Description:

Base Template: <None>

Type:

Categories:

Default Attrib

Naming Pattern:

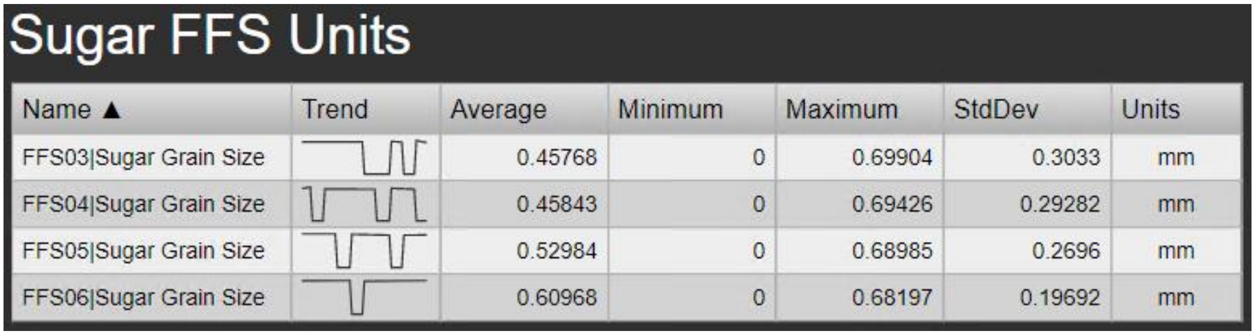
□ Allow Extensions □ Base Template Only

Figure 8. Disable Allow Extensions

Step 9. Important: Click on Check In **to save your changes.**

Step 10. Move to PI Vision → display **Sandbox_Overview_FFS**

Step 11. Refresh the page and verify that now the table shows different trends and values for each Sugar Grain Size attribute.



Sugar FFS Units

Name ▲	Trend	Average	Minimum	Maximum	StdDev	Units
FFS03|Sugar Grain Size		0.45768	0	0.69904	0.3033	mm
FFS04|Sugar Grain Size		0.45843	0	0.69426	0.29282	mm
FFS05|Sugar Grain Size		0.52984	0	0.68985	0.2696	mm
FFS06|Sugar Grain Size		0.60968	0	0.68197	0.19692	mm

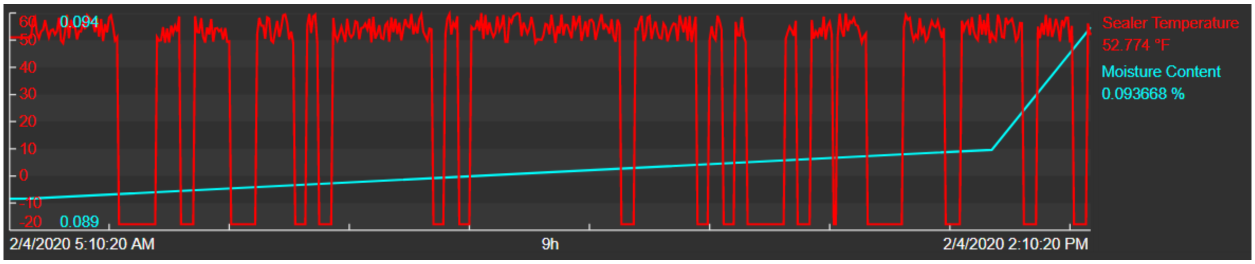
✓ Best Practice	Always use AF templates. If one or more assets differ from the others, you can leverage **Derived Templates** (and/or the Excluded attribute property) to handle these variations. We recommend caution when using **Allow Extensions**, as the additional attributes created with this option must be manually configured for each “extended” element.

2.4.3. Units of Measure

The UOM database provides automatic handling of simple conversions between units of measure for attributes of the same UOM class. A UOM class is defined by the fundamental dimensions of its measurement. Examples of UOM classes are Mass, Volume, and Density. The UOM database comes preloaded with numerous standard unit-of-measure classes and conversion factors. You can extend these classes by adding new units of measure, as well as new measurement classes. The implementation of UOM is based on the International System of Units (SI).

While navigating in PI Vision, we noticed that the Sealer Temperature was out of the expected range (it should be between 70 and 270 ºF).

Issue # 3. The attribute Sealer temperature shows values outside the expected range.



Since this issue occurs for all FFS units, it's not likely due to a sensor's malfunctioning! Let's verify that the AF attributes have been correctly configured.

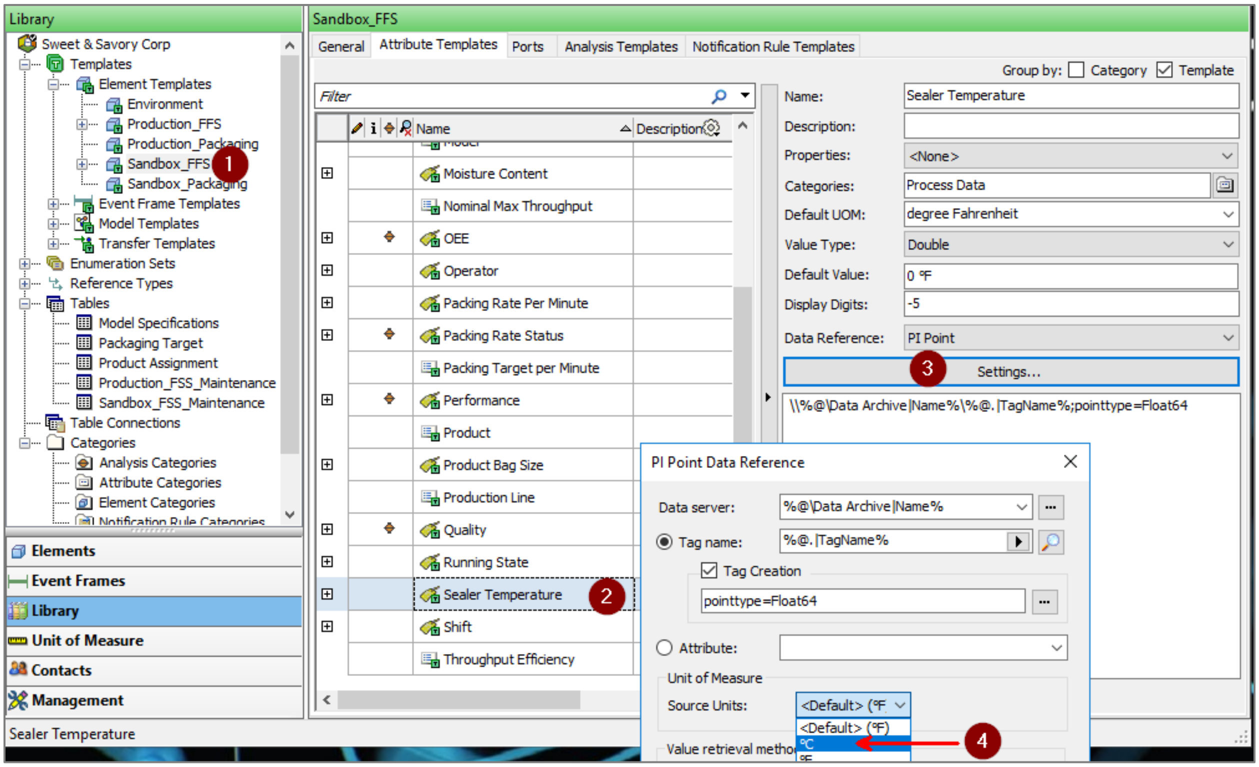
Sweet & Savory Corp has bought a stock of temperature sensors from Europe, so **the temperature is actually measured in degrees Celsius**. However, we expect the AF Server to provide us the values in ºF while handling the UOM conversion behind the scenes. Let's make sure that AF is aware of the UOM of the source data (Figure 9).

Step 1. In PSE, move to the Library tab → **Sandbox_FFS** → Attribute Templates tab

Step 2. Select the **Sealer Temperature**.

Step 3. Click on **Settings…**

Step 4. Select explicitly **ºC** instead of the default value (in this case *<Default> (ºF)*).



Name: Sealer Temperature
Description:
Properties: <None>
Categories: Process Data
Default UOM: degree Fahrenheit
Value Type: Double
Default Value: 0 ºF
Display Digits: -5
Data Reference: PI Point

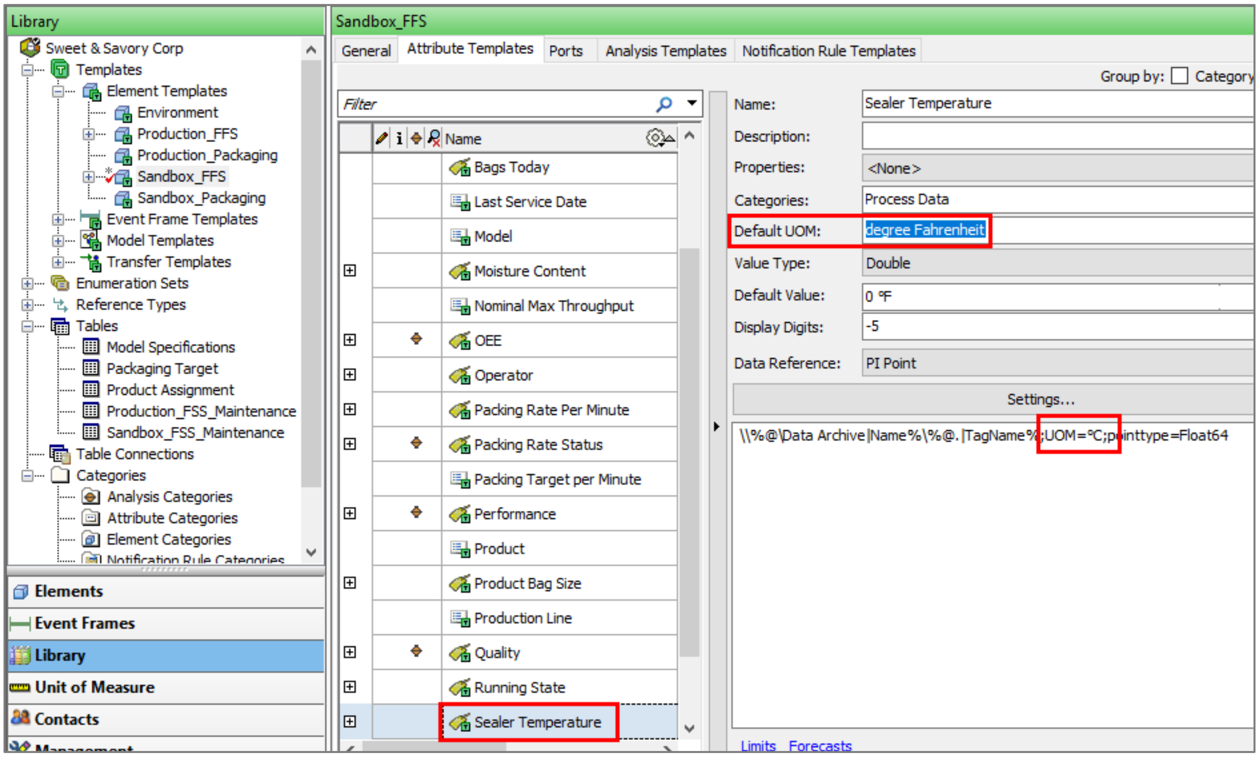
\\%@\Data Archive|Name%\%@.|TagName%;pointtype=Float64

PI Point Data Reference

Data server: %@\Data Archive|Name%
● Tag name: %@.|TagName%
☑ Tag Creation
pointtype=Float64
○ Attribute:
Unit of Measure
Source Units: <Default> (ºF
<Default> (ºF)
ºC
ºF
Value retrieval metho

Figure 9. Setting the Source UOM.

Step 5. Click OK and verify that the **Default UOM** is still ºF (Figure 10). The **Default UOM** configuration field defines the unit of measure used **to display the attribute's values on client applications**.



Name: Sealer Temperature
Description:
Properties: <None>
Categories: Process Data
Default UOM: degree Fahrenheit
Value Type: Double
Default Value: 0 °F
Display Digits: -5
Data Reference: PI Point

```
\\%@\Data Archive|Name%\%@.|TagName%;UOM=°C;pointtype=Float64
```

Figure 10. Default UOM vs Source Unit

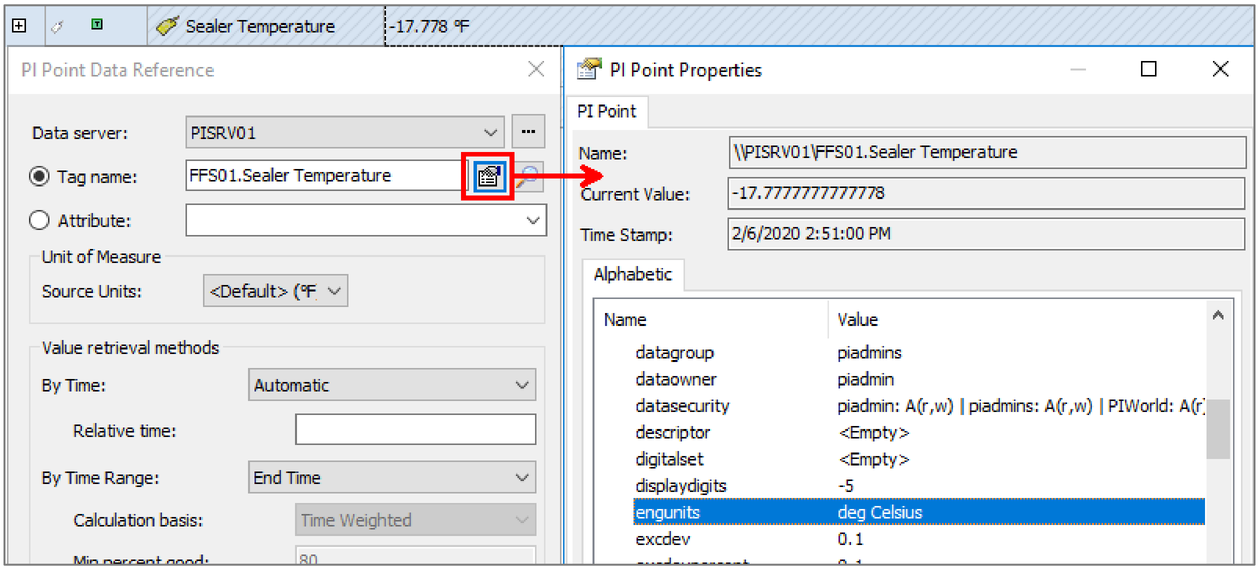
Step 6. Go back to PI Vision, refresh the display and verify that now the values fall within the expected operational range.

✓ Best Practice	Make sure you always assign a unit of measure (UOM) rather than just keeping the default one. This prevents that the values archived on the Data Archive are misinterpreted in case the Display UOM of the AF attribute is changed later in time.

You may be wondering… and what about the Data Archive? Aren’t UOMs defined for each PI Point as well?

NERD ALERT – Data Archive engineering units vs Asset Framework UOMs

You may run into a situation where you need to set the UOM of an AF attribute and you want to inspect the settings for the PI Point. The easiest way to do this is to select the attribute within an element and click the “PI Point Properties” button (Figure 11).



Sealer Temperature -17.778 °F

Data server: PISRV01

● Tag name: FFS01.Sealer Temperature

○ Attribute:

Unit of Measure

Source Units: <Default> (°F

Value retrieval methods

By Time: Automatic

Relative time:

By Time Range: End Time

Calculation basis: Time Weighted

Min percent good: 80

Name: \\PISRV01\FFS01.Sealer Temperature

Current Value: -17.7777777777778

Time Stamp: 2/6/2020 2:51:00 PM

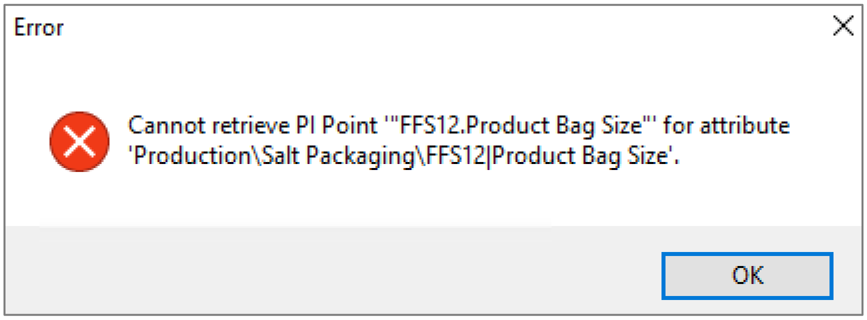
Name	Value
datagroup	piadmins
dataowner	piadmin
datasecurity	piadmin: A(r,w) | piadmins: A(r,w) | PIWorld: A(r
descriptor	<Empty>
digitalset	<Empty>
displaydigits	-5
engunits	deg Celsius
excdev	0.1

Figure 11. The PI Point Properties window

Remember: **the *engunit* field is optional** when configuring a PI Point (just as the UOMs are optional for AF attributes), so you might not find the information you are looking for. Also, the *engunit* is **simply a character string**, it is not involved in UOM conversion and it does not respect the UOM terminology used in the AF Server (degrees Celsius can be expressed as ºC, or as deg C, or in any other way). It is discretion of the Data Archive administrator to correctly configure the *engunit* field of the PI Points.

A common situation you may run into is that the attribute is resolving the PI Point using a replacement character string such as `%@.|TagName%`. If this is the case, when clicking on the PI Point properties button you'll probably get an error popup such as:

`Cannot retrieve PI Point '<PI Point name>' for attribute '<path to AF attribute>'.`



Error

Cannot retrieve PI Point '"FFS12.Product Bag Size"' for attribute 'Production\Salt Packaging\FFS12|Product Bag Size'.

Before we can view the properties of that PI Point, we need to force AF to resolve the PI Point name and store it in the attribute configuration. This can be done by right-clicking on the attribute and selecting "*Create or Update PI Point*".

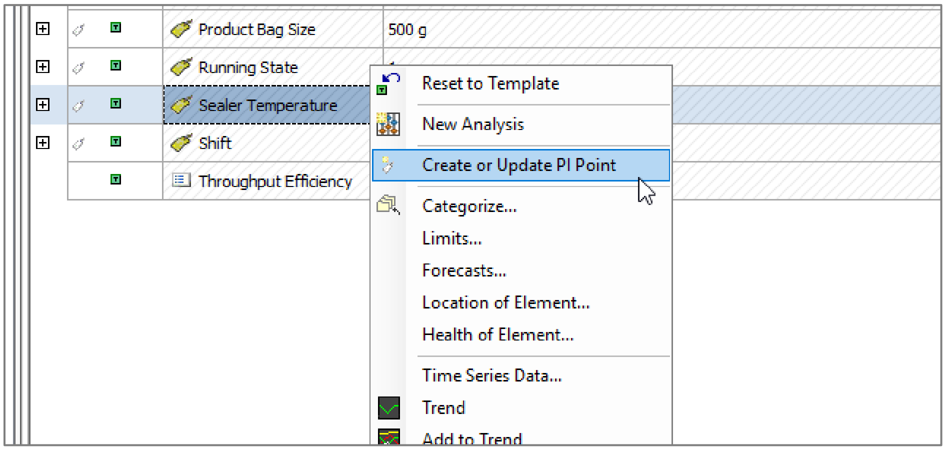
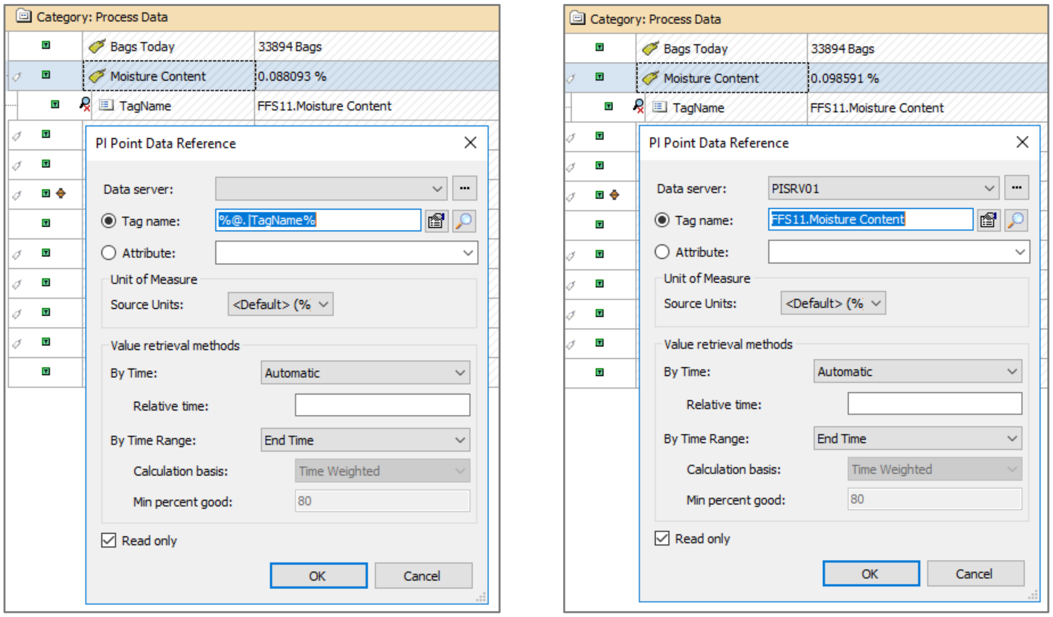


Figure 12 shows the same AF attribute, before (left) and after (right) performing the "Create or Update PI Point" operation.



Category: Process Data

Bags Today	33894 Bags
Moisture Content	0.088093 %
TagName	FFS11.Moisture Content

PI Point Data Reference

Data server:

● Tag name: %@.|TagName%

○ Attribute:

Unit of Measure

Source Units: <Default> (%

Value retrieval methods

By Time: Automatic

Relative time:

By Time Range: End Time

Calculation basis: Time Weighted

Min percent good: 80

☑ Read only

Category: Process Data

Bags Today	33894 Bags
Moisture Content	0.098591 %
TagName	FFS11.Moisture Content

PI Point Data Reference

Data server: PISRV01

● Tag name: FFS11.Moisture Content

○ Attribute:

Unit of Measure

Source Units: <Default> (%

Value retrieval methods

By Time: Automatic

Relative time:

By Time Range: End Time

Calculation basis: Time Weighted

Min percent good: 80

☑ Read only

Figure 12. Tag name resolution before and after the “Create or Update PI Point” operation.

When this action is carried out, AF resolves the replacement characters and locks in the settings. This is often done to improve performance in larger systems as the client tools will no longer have to use processing power or network bandwidth performing the conversion from a replacement character string to the actual tag name. This efficiency improvement is negligible on a small system but becomes relevant for large, corporate-level systems. The drawback is that, in case the value of attribute *TagName* changes, the PI Point data reference attribute does not get automatically updated. The solution in this case is resetting the attribute configuration to the template, and then performing again the “Create or Update PI Point” operation.

For more information on what the “Create or Update PI Point” operation does, please read Attribute indicators for updates of PI point data references.

2.4.4. AF Tables

Tables are held in the AF database to provide contextual information through the Table Lookup data reference. Tables can provide information about the equipment or process entities; also, they can be used to store tag names or other configuration information.

Tables can be created internally, imported from an external relational or tabular data source, or linked dynamically to an external relational data source. In this way, Tables can expose information in maintenance, production planning, or equipment databases for use by PI client applications.

When you opened PSE for the first time and navigated to the sandbox elements, you probably noticed that it took a long time to populate the attributes' values in the Attributes tab. We will see that this slowness is caused by a single heavy hitter, the attribute **Last Service Date**. This attribute is a Table Lookup, which means that it… looks up a value in a table, using a SQL-like query of type:

SELECT *column* FROM *table* WHERE *whereClause* ORDER BY *column* ASC|DESC;

Let's look at the table we use to populate the Last Service Date attribute:

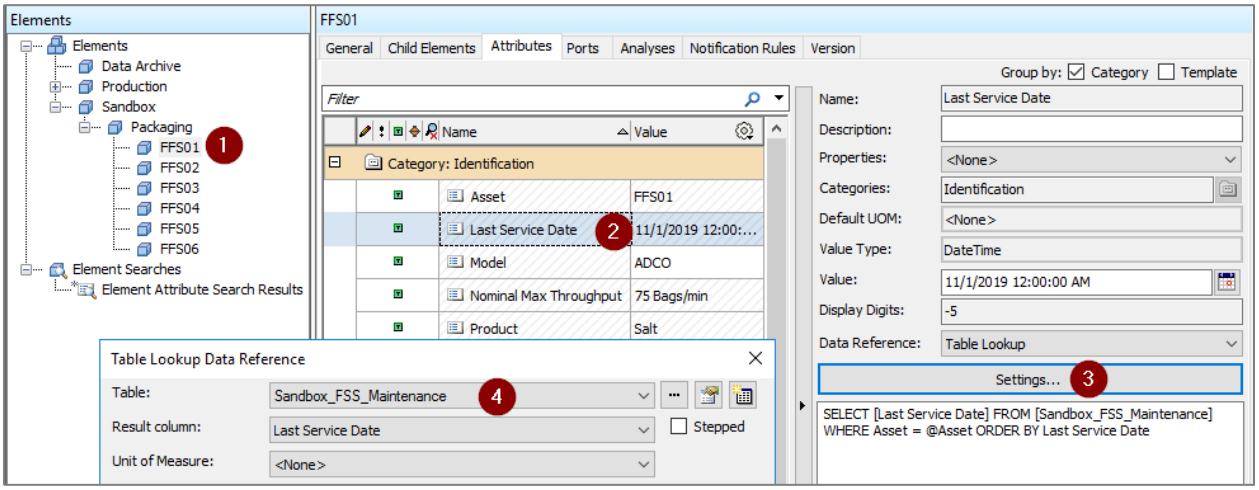
Step 1. In PSE, move to the Elements tab → Sandbox → Packaging → FFS01.

Step 2. Open the Attributes tab and select the Last Service Date attribute.

Step 3. Click on **Settings…**

Step 4. Take note of the table used (this table appears also in the SQL query under the Settings button)

SELECT [Last Service Date] FROM [**Sandbox_FFS_Maintenance**] WHERE Asset = @Asset ORDER BY Last Service Date



Name	Value
Asset	FFS01
Last Service Date	11/1/2019 12:00:…
Model	ADCO
Nominal Max Throughput	75 Bags/min
Product	Salt

Name: Last Service Date
Description:
Properties: <None>
Categories: Identification
Default UOM: <None>
Value Type: DateTime
Value: 11/1/2019 12:00:00 AM
Display Digits: -5
Data Reference: Table Lookup

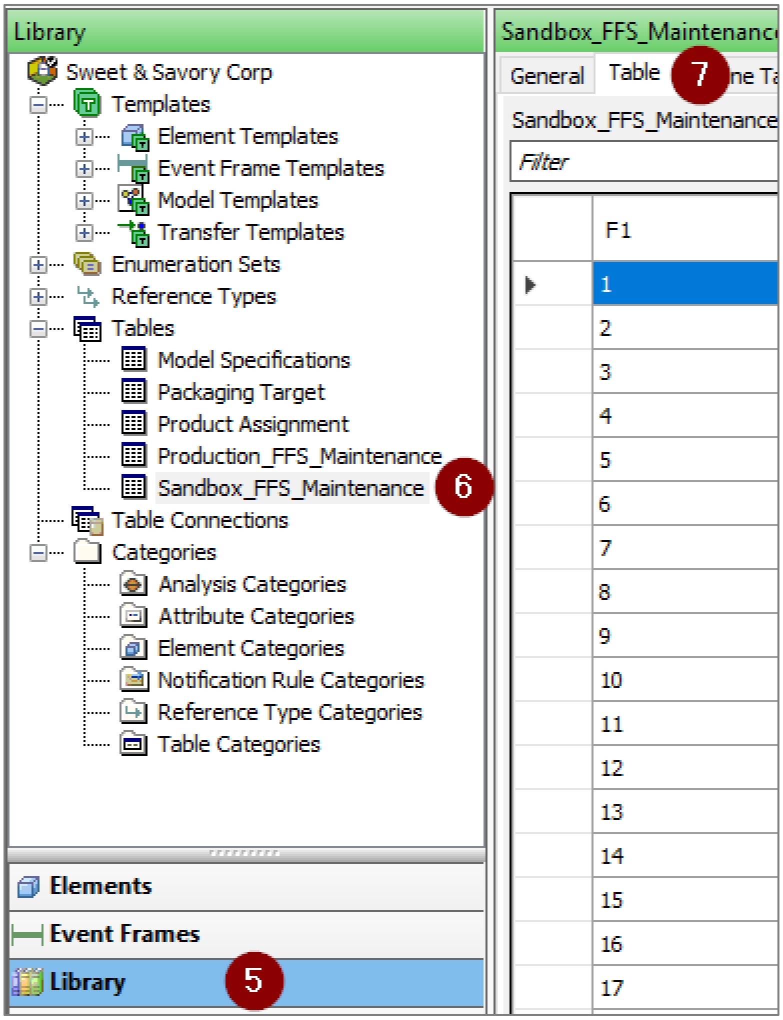
SELECT [Last Service Date] FROM [Sandbox_FSS_Maintenance] WHERE Asset = @Asset ORDER BY Last Service Date

Table: Sandbox_FSS_Maintenance
Result column: Last Service Date
Unit of Measure: <None>

Step 5. Move to the Library tab and open the Tables dropdown list.

Step 6. Select table **Sandbox_FFS_Maintenance.**

Step 7. Move to the Table tab.



How many rows are present in this table? *[hint: far too many!]*

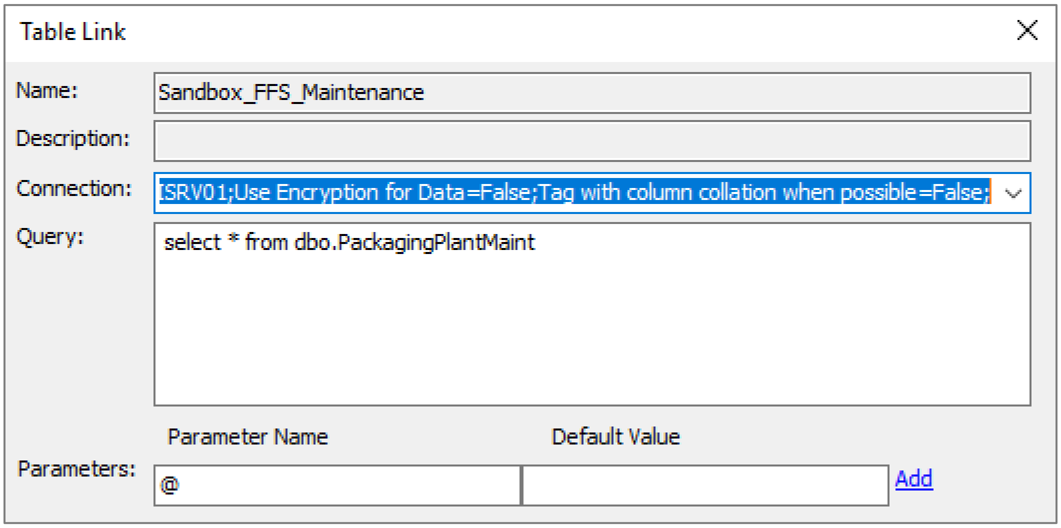
Do we need all these rows? *[hint: no, we only need the FFS Units that appear in our database!]*

The table Sandbox_FFS_Maintenance is stored in a central SQL Server at the corporate level and contains asset metadata from all the production sites around the world (around 300 000 FFS units… probably unrealistic, but good for our discussion!). For our goals, we only need to retrieve the data for our pilot site. This has already been done correctly in the Production environment, so we just need to understand how it works and replicate it in the sandbox environment.

Step 8. Still on table Sandbox_FFS_Maintenance, move to the **General** tab.

Step 9. Click on the "**Link…**" button. You'll see a Connection String and a query to an external SQL Server. The current query is retrieving all rows and columns from the external SQL table:

```
select * from dbo.PackagingPlantMaint
```



Name: Sandbox_FFS_Maintenance

Description:

Connection: ISRV01;Use Encryption for Data=False;Tag with column collation when possible=False;

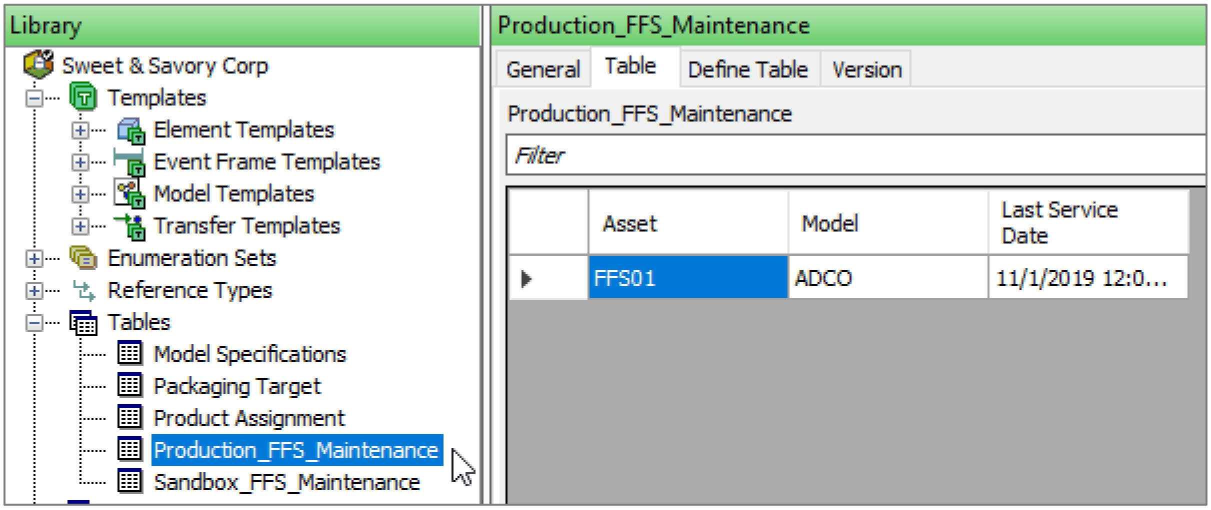
Query: select * from dbo.PackagingPlantMaint

Parameter Name	Default Value
@	

Step 10. Click on **Cancel** and select table **Production_FFS_Maintenance**.

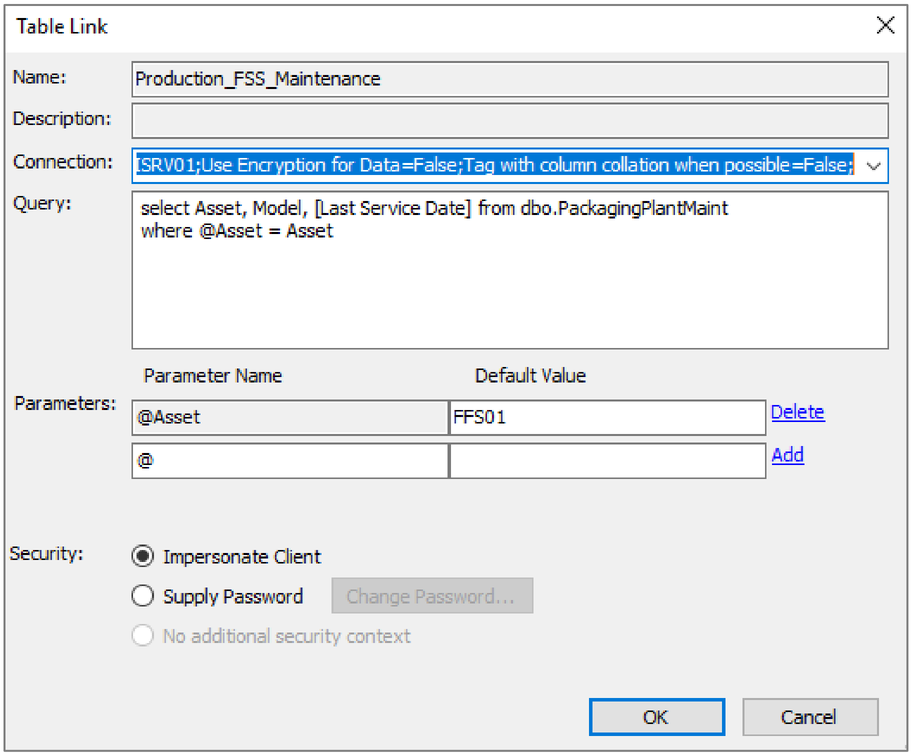
Step 11. Move to the Table tab, how many rows do you see?

NB: The Table tab shows only one row corresponding to the default value of the @Asset parameter (FFS01), but the query will work fine for any other AF element.



Asset	Model	Last Service Date
FFS01	ADCO	11/1/2019 12:0...

Step 12. Go to General → "**Link…**" and check how the query is performed. You'll notice that the SQL query contains a parameter (*where @Asset = Asset*). You can hit Cancel to close the window.



Name: Production_FSS_Maintenance

Description:

Connection: ISRV01;Use Encryption for Data=False;Tag with column collation when possible=False;

Query:

```
select Asset, Model, [Last Service Date] from dbo.PackagingPlantMaint
where @Asset = Asset
```

Parameter Name	Default Value
@Asset	FFS01
@	

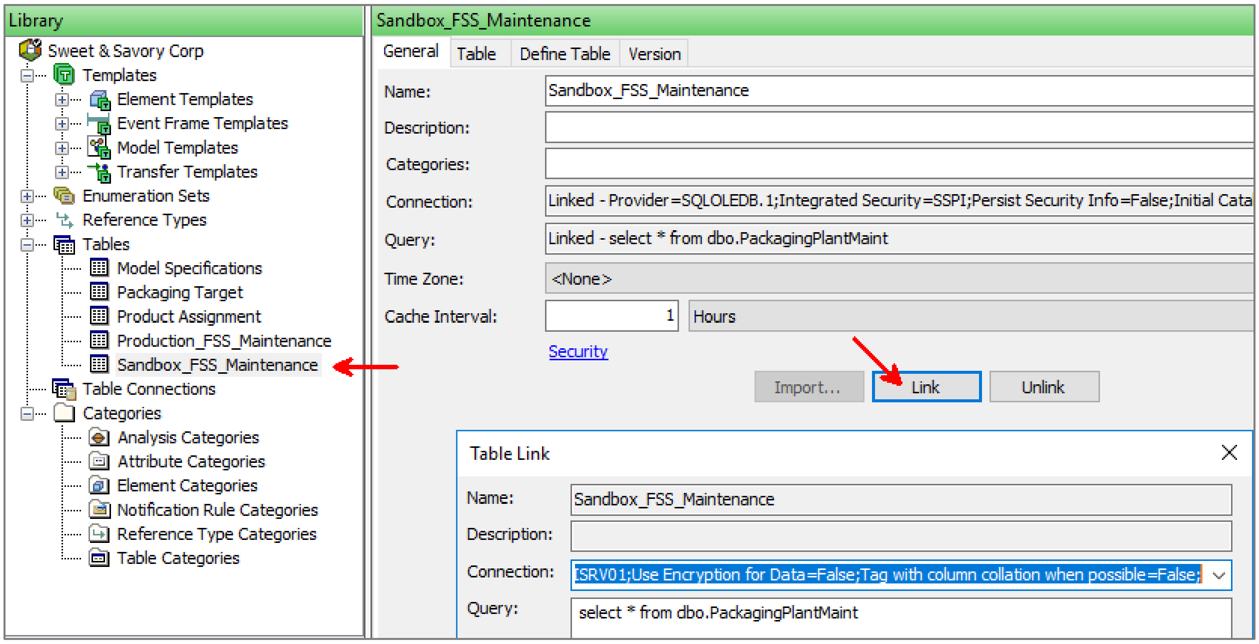
Security:
- ● Impersonate Client
- ○ Supply Password
- ○ No additional security context

Using parameters in a linked table query is useful to limit the number of rows returned from a very large external table. You can add conditions and parameters to return more targeted results, such as all rows that include a device or manufacturer ID number, specific for each table lookup data reference. (LiveLibrary).

We will now go back to the Sandbox table and edit the query so that only needed information is returned (Figure 13):

Step 13. Open table **Sandbox_FFS_Maintenance** → General tab.

Step 14. Select the **Link…** button.



Name: Sandbox_FSS_Maintenance

Description:

Categories:

Connection: Linked - Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Cata

Query: Linked - select * from dbo.PackagingPlantMaint

Time Zone: <None>

Cache Interval: 1 Hours

Table Link

Name: Sandbox_FSS_Maintenance

Description:

Connection: ISRV01;Use Encryption for Data=False;Tag with column collation when possible=False;

Query: select * from dbo.PackagingPlantMaint

Figure 13. Link configuration tab of table Sandbox_FSS_Maintenance

Step 15. Edit the query as follows (feel free to copy and paste from the .PDF file of this workbook):

```
Select * from dbo.PackagingPlantMaint
Where @Asset = Asset
```

Step 16. In the Parameters Section, add the parameter @Asset (see also Figure 14):

Parameter Name	Default Value
@Asset	FFS01

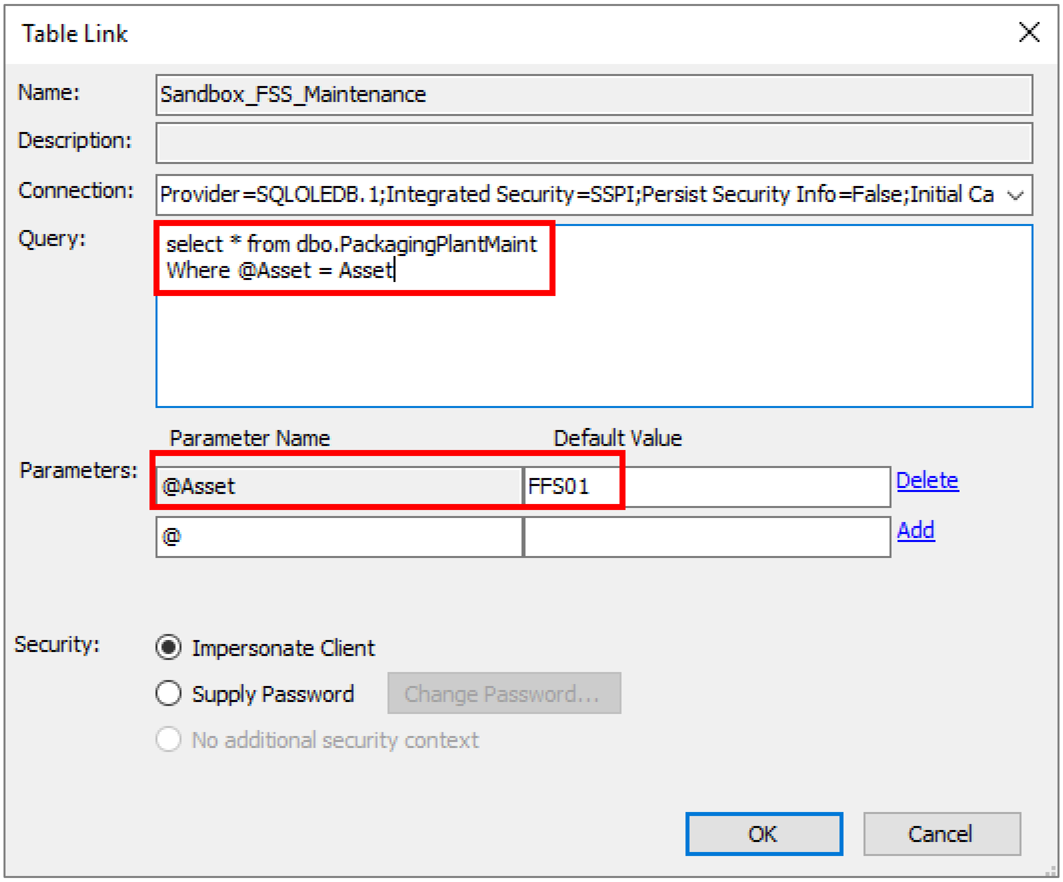


Table Link

Name: Sandbox_FSS_Maintenance

Description:

Connection: Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Ca

Query:

```
select * from dbo.PackagingPlantMaint
Where @Asset = Asset
```

Parameters:

Parameter Name	Default Value	
@Asset	FFS01	Delete
@		Add

Security:

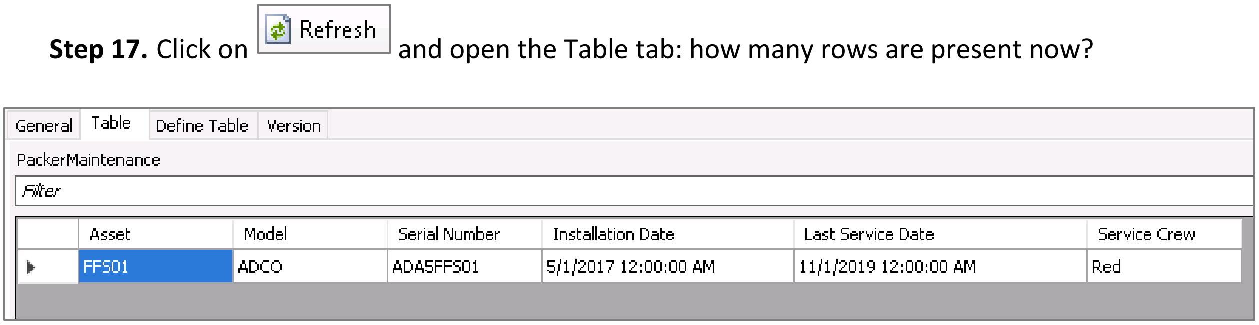
● Impersonate Client

○ Supply Password

○ No additional security context

Figure 14. Modifications to the Sandbox_FFS_Maintenance table

Step 17. Click on Refresh and open the Table tab: how many rows are present now?



PackerMaintenance

Filter

Asset	Model	Serial Number	Installation Date	Last Service Date	Service Crew
FFS01	ADCO	ADA5FFS01	5/1/2017 12:00:00 AM	11/1/2019 12:00:00 AM	Red

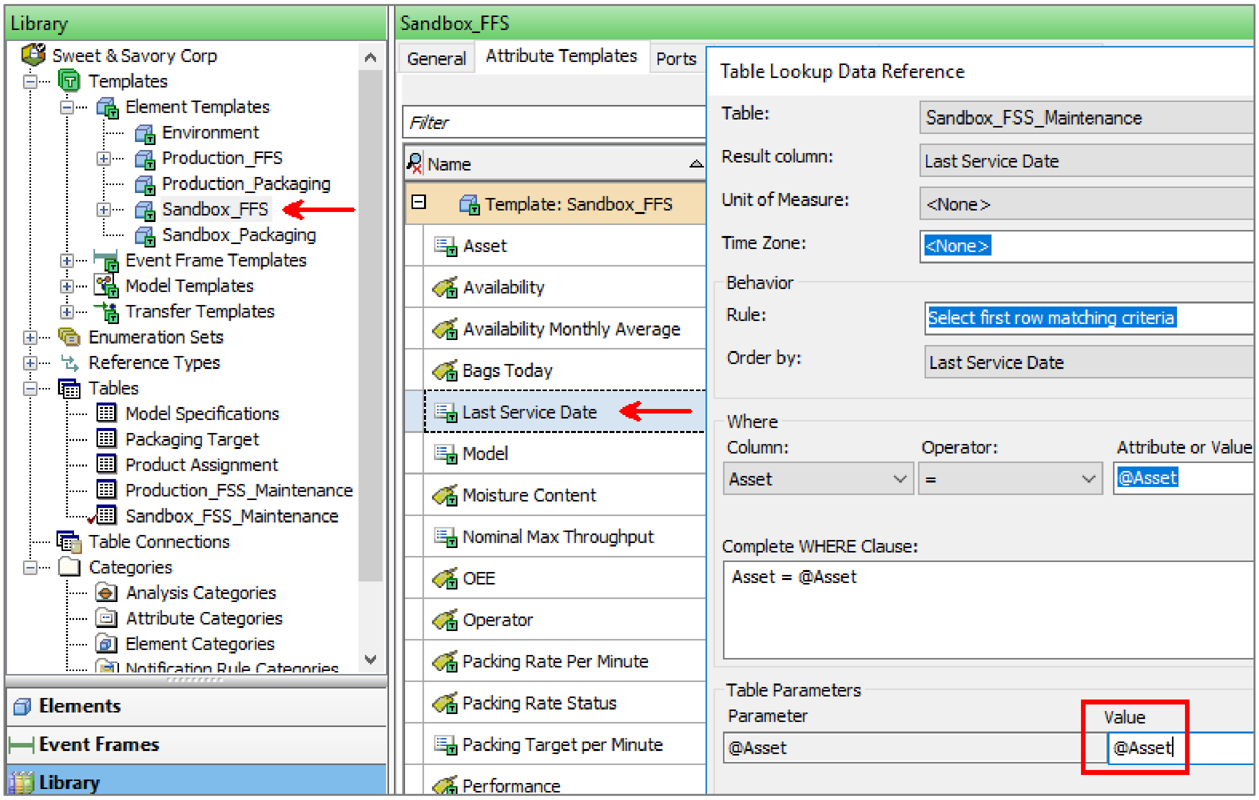
Only one row is returned, and it corresponds to the Default value of the SQL query.

NB: as stated before, the parameter value in the configuration above is just an example parameter value (a default value) and it doesn't affect the query when evaluated for a different AF element.

Step 18. Move to the **Sandbox_FFS** and select the Attribute Templates tab.

Step 19. Select the **Last Service Date** and click on Settings...

Step 20. Add the Table Parameter as in the screenshot below (Parameter @Asset = Value **@Asset**).



Library

- Sweet & Savory Corp
  - Templates
    - Element Templates
      - Environment
      - Production_FFS
      - Production_Packaging
      - Sandbox_FFS
      - Sandbox_Packaging
    - Event Frame Templates
    - Model Templates
    - Transfer Templates
  - Enumeration Sets
  - Reference Types
  - Tables
    - Model Specifications
    - Packaging Target
    - Product Assignment
    - Production_FFS_Maintenance
    - Sandbox_FSS_Maintenance
  - Table Connections
  - Categories
    - Analysis Categories
    - Attribute Categories
    - Element Categories
    - Notification Rule Categories

Elements | Event Frames | Library

Sandbox_FFS

General | Attribute Templates | Ports

Filter

Name

- Template: Sandbox_FFS
  - Asset
  - Availability
  - Availability Monthly Average
  - Bags Today
  - Last Service Date
  - Model
  - Moisture Content
  - Nominal Max Throughput
  - OEE
  - Operator
  - Packing Rate Per Minute
  - Packing Rate Status
  - Packing Target per Minute
  - Performance

Table Lookup Data Reference

Table: Sandbox_FSS_Maintenance

Result column: Last Service Date

Unit of Measure: <None>

Time Zone: <None>

Behavior

Rule: Select first row matching criteria

Order by: Last Service Date

Where

Column:	Operator:	Attribute or Value
Asset	=	@Asset

Complete WHERE Clause:

Asset = @Asset

Table Parameters

Parameter	Value
@Asset	@Asset

In this way, every time a value is requested by a Table Lookup attribute, the parametrized query will be passed to the external SQL server and only the requested value will be returned.

Step 21. Important: Click on Check In **to save your changes.**

Step 22. In order to verify whether the caching performance has improved, close PI System Explorer and PI Vision. Then reopen PI System Explorer and move between AF elements under the Sandbox branch (make sure you have the Attributes tab open). Do you notice any improvements in terms of data retrieval speed? [*hint: yes! :D*]

If you like, you can further limit the number of results returned by the SQL Query. What we need from the external SQL table are just the Asset name and the Last Service Date. We don't need to import also the Model, Serial Number, Installation Date and Service Crew. We can modify the query of the **Sandbox_FFS_Maintenance** table as follows:

```
select Asset, [Last Service Date] from dbo.PackagingPlantMaint
where @Asset = Asset
```

NB: Don't forget the **square brackets** around Last Service Date. The space characters in the column name Last Service Date would break the SQL query unless square brackets are used.

Still hungry for knowledge on AF Tables? For more information on parametrized queries, please visit Data references from outside the PI System and Parameters for linked table queries.

Also, this article covers how AF Tables handle local caching: How AFTable Caching Works (4).

Last but not least, this PI Square post illustrates good and bad uses of Table Lookup, Formula and Analysis data references: Asset Analytics Best Practices - Part 3: Input Attributes (5).

✓ Best Practice	Don't bring data you don't need into the PI System. In the case of table lookups, you can use parametrized queries to only fetch data that will be used. A rule-of-thumb is to avoid importing tables with over 10 000 rows.

Before we move on, let's look at what attributes could slow down data retrieval, and what actions you can take to solve these performance issues.

NERD ALERT – Heavy hitters

Table Lookup

Table Lookup attributes that point to externally linked tables are the usual suspects when you experience slow data retrieval. The performance issue can be caused by the size of the table (as in our example), by network latencies between the Asset Framework server and the external data source, by an overloaded external data source etc.

As you see, not all these issues can be directly addressed in Asset Framework. Parametrized queries can drastically reduce latency when the main issue is the table size; another option is importing the table in AF (in this case, it is a good practice to limit your imported tables to 10,000 rows of data or less).

Formula

Formula data reference allows you to perform simple calculations that are evaluated on-the-fly by the client application (e.g. PI Vision, PI DataLink). Performance issues can arise in case of long dependency chains (i.e. when a Formula needs to wait for its input values coming from a chain of other Formulas).

The best solution is to move the calculation to Asset Analytics and use PI Points to store the calculation results.

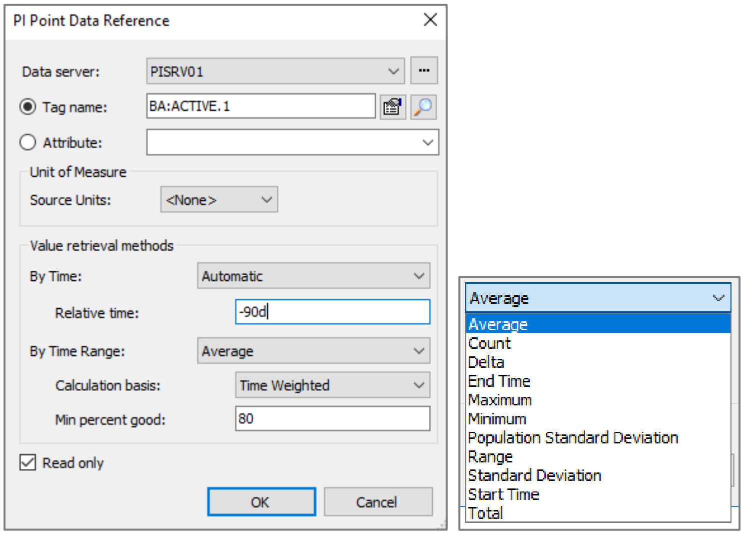
Summary calculation via Value Retrieval Methods

Value Retrieval Methods, available for the PI Point data reference, allow you to perform statistical calculations (such as an average or a total) over a desired time range. This calculation is requested on-the-fly by the client application and performed by the PI Archive Subsystem on the Data Archive.

In the example in Figure 15, we request a 90-day average for PI Point BA:ACTIVE.1. The evaluation time will depend on the data density (the number of events for that PI Point and for the requested time range) and the available resources of the Data Archive.

Again, the best solution is to move the calculation to Asset Analytics and use a PI Point to store the calculation results.

For more information on Value Retrieval Methods, please refer to [Configuration of retrieval methods for attribute values]().



PI Point Data Reference

Data server: PISRV01

● Tag name: BA:ACTIVE.1

○ Attribute:

Unit of Measure

Source Units: <None>

Value retrieval methods

By Time: Automatic

Relative time: -90d

By Time Range: Average

Calculation basis: Time Weighted

Min percent good: 80

☑ Read only

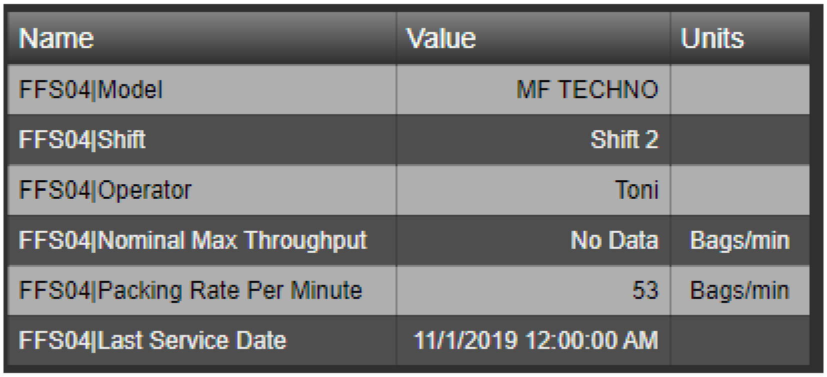
Figure 15. Value Retrieval Methods for the PI Point data reference (left) and available functions (right)

2.4.5. Enumeration Sets

An enumeration set is an ordinal list of sequential integer values, which are mapped to names. It allows AF attributes to refer to a common term instead of a number. You typically use enumeration sets to establish predefined values for attribute templates. When you configure AF attributes based on those templates, you have users select those values from pre-populated lists rather than typing values manually. This helps ensure you have consistent nomenclature throughout your database.

Remember what happened when you opened **Sandbox_FFS_Monitoring** and switched to FFS04?

Issue # 5. Asset FFS04 shows "No Data" for the Nominal Max Throughput



Name	Value	Units
FFS04|Model	MF TECHNO	
FFS04|Shift	Shift 2	
FFS04|Operator	Toni	
FFS04|Nominal Max Throughput	No Data	Bags/min
FFS04|Packing Rate Per Minute	53	Bags/min
FFS04|Last Service Date	11/1/2019 12:00:00 AM	

Let's see where this issue leads us to!

Step 1. In PI System Explorer, under the **Sandbox** branch select **FFS04**.

Step 2. Select attribute **Nominal Max Throughput** and click on **Settings…** As we saw in exercise 2.4.3, this attribute is of type Table Lookup, and retrieves its value using a SQL-like query.

Attribute **Model** is used in the configuration of attribute **Max Throughput** as input parameter of the WHERE clause:

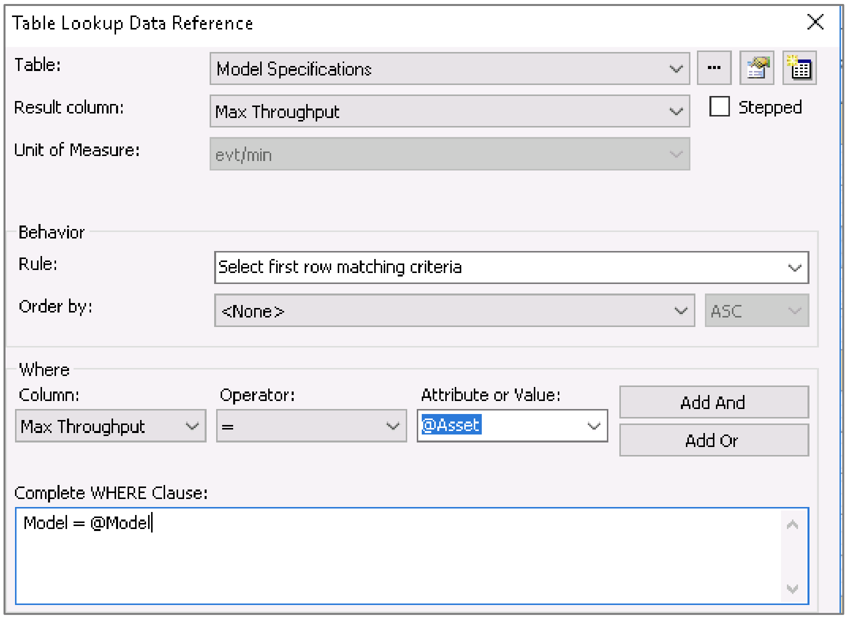


Table Lookup Data Reference

Table: Model Specifications

Result column: Max Throughput

□ Stepped

Unit of Measure: evt/min

Behavior

Rule: Select first row matching criteria

Order by: <None> ASC

Where

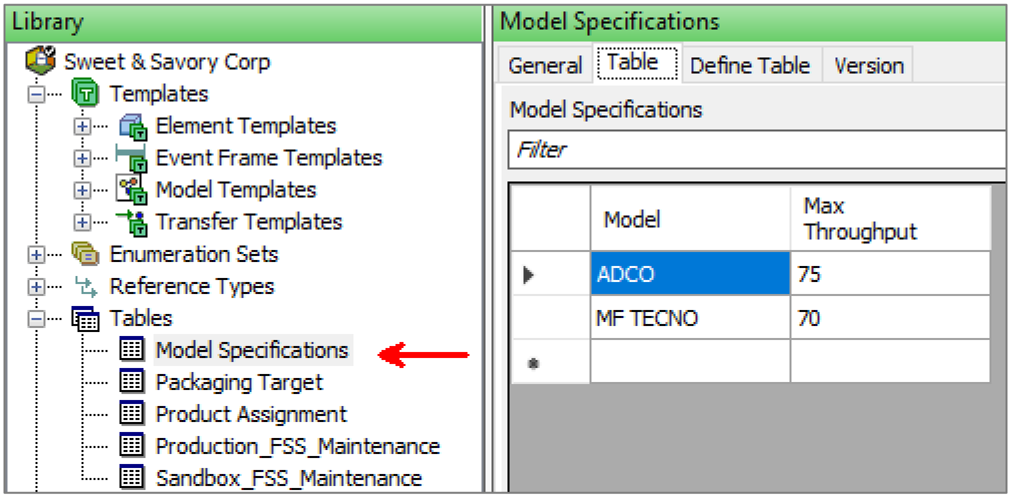
Column:	Operator:	Attribute or Value:
Max Throughput	=	@Asset

Complete WHERE Clause:

Model = @Model

Let’s look at the AF Table that this attribute retrieves its value from.

Step 3. Move to Library → **Tables** → **Model Specifications** → Table tab

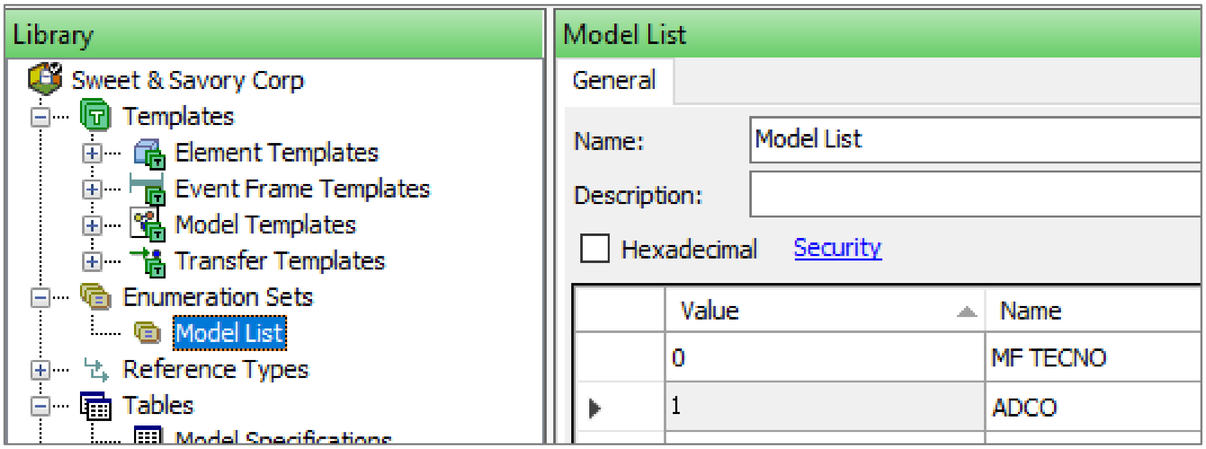


Model Specifications

Model	Max Throughput
ADCO	75
MF TECNO	70

We notice that the two models are ADCO and MF TECNO, while the value entered in the Model attribute for FFS04 is MF TECHNO. The extra “**H**” causes the Nominal Max Throughput attribute to fail to resolve, and the Throughput Efficiency fails as well, since it uses the Nominal Max Throughput as a formula input.

In order to solve this issue and to **avoid future typos**, we need to edit the template **Sandbox_FFS** so that the attribute **Model** uses an Enumeration Set. We want to use the Enumeration Set **Model**.



Name: Model List

Description:

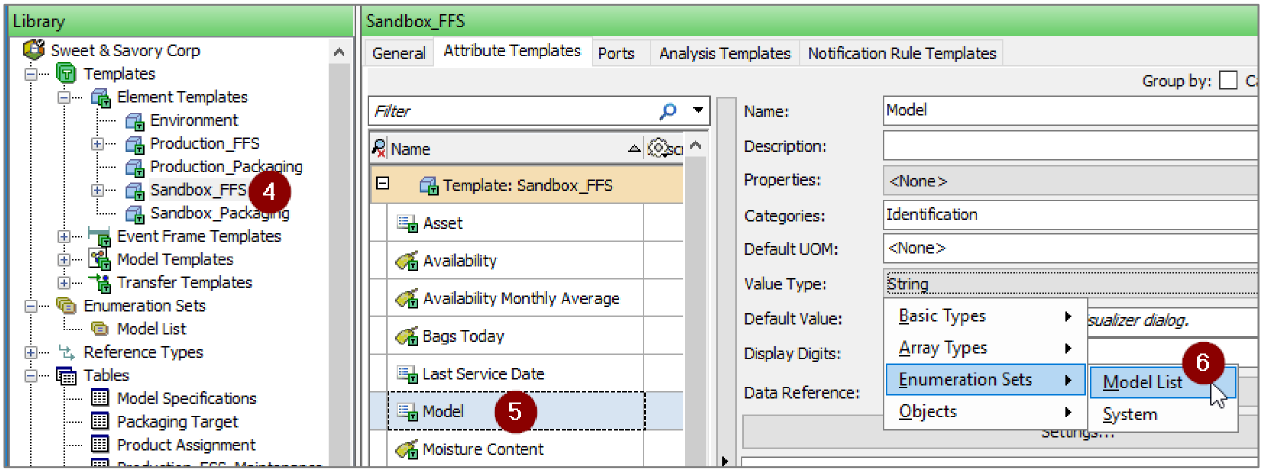
□ Hexadecimal Security

Value	Name
0	MF TECNO
1	ADCO

Step 4. Go to the **Library** tab and select **Element Templates → Sandbox_FFS**.

Step 5. Move to the Attribute Templates tab and select the attribute Model.

Step 6. Edit the Value Type and choose **Enumeration Sets → Model**.



Name: Model

Description:

Properties: <None>

Categories: Identification

Default UOM: <None>

Value Type: String

Default Value: ...sualizer dialog.

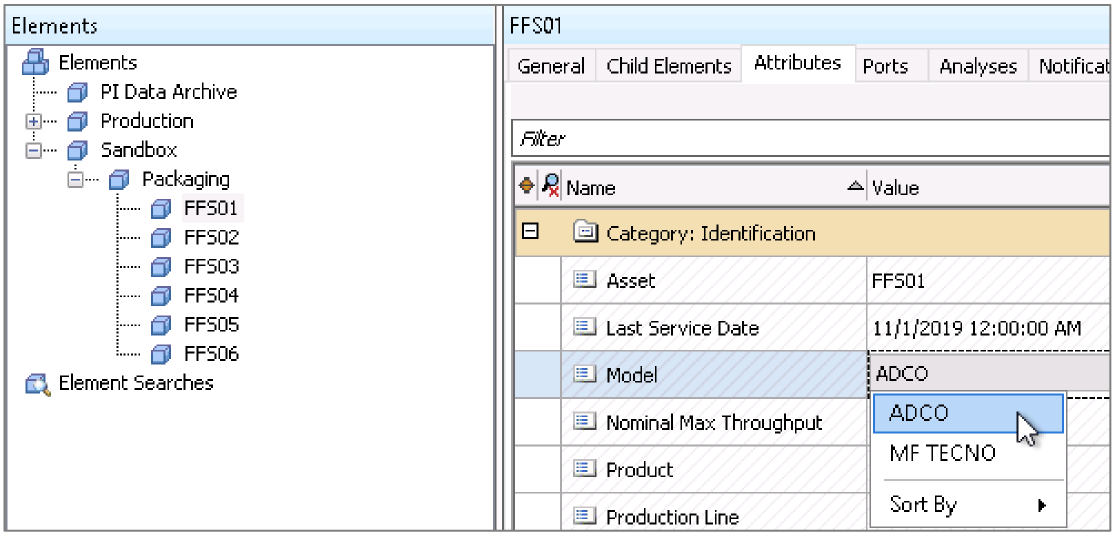
Display Digits:

Data Reference:

Step 7. Move to the Elements tab and specify the model of your FFS units under the Sandbox environment. Refer to Table 3 (next page) to choose the correct model for each element.

Step 8. Important: Click on Check In **to save your changes.**

Step 9. Move to PI Vision → display **Sandbox_FFS_Monitoring**, refresh the page and verify that now the table shows good values for the FFS04 attribute Nominal Max Throughput.



Name	Value
Category: Identification	
Asset	FFS01
Last Service Date	11/1/2019 12:00:00 AM
Model	ADCO
Nominal Max Throughput	
Product	
Production Line	

Table 3. Models associated with FFS units 1 to 6:

FFS unit	Model
FFS01	ADCO
FFS02	ADCO
FFS03	ADCO
FFS04	MF TECNO
FFS05	ADCO
FFS06	MF TECNO

Note: Starting with **PI Server 2018 SP2**, **Enumeration Sets can be created from Digital Sets** (stored in the Data Archive) using PI System Explorer. This helps you ensure consistency between the Data Archive and the Asset Framework server. For more information, refer to Create enumeration sets from digital state sets and review digital state sets on a PI Data Archive.

✓ Best Practice	Use Enumeration Sets to establish predefined values for attribute templates. This ensures you have consistent nomenclature throughout your database.

2.4.6. NERD ALERT - String Builder

The String Builder data reference enables you to apply string manipulation functions, such as concatenation, to attributes' values, and output a reformatted string. This is useful when you need to obtain a string or numeric value type from other element attributes.

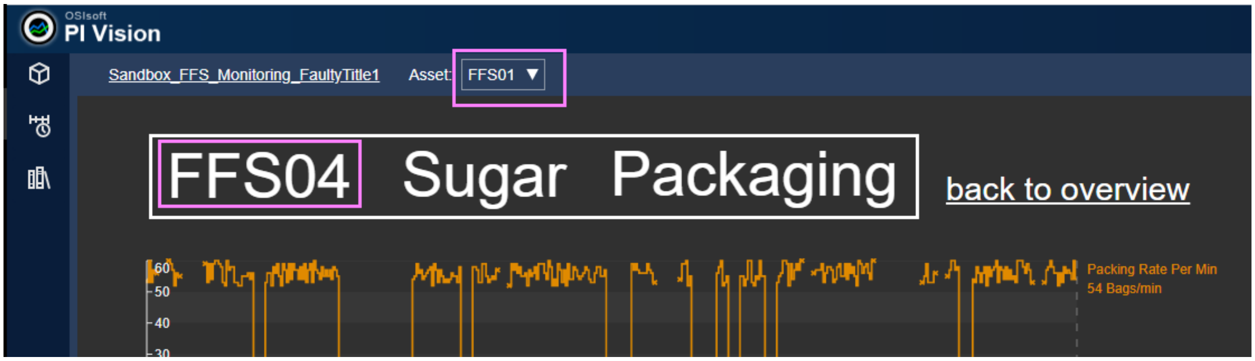
Below are some applications of the String Builder data reference:

- ✓ Concatenate strings (Element Name + Attribute Name + Value).
- ✓ Format dates and numbers.
- ✓ Build paths to elements and attributes.
- ✓ Parse comments from operators stored in PI Points.
- ✓ Display element information as an attribute.

For more information please visit: String Builder data references.

Let's now see it in action. Remember the last issue we observed in section 2.3?

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset



The title "FFS04 Sugar Packaging" was built using static text boxes, hence it doesn't update when changing asset. By hovering over any of the symbols we see that the process data has updated with the correct asset (e.g. FFS01), but the title still shows FFS04.

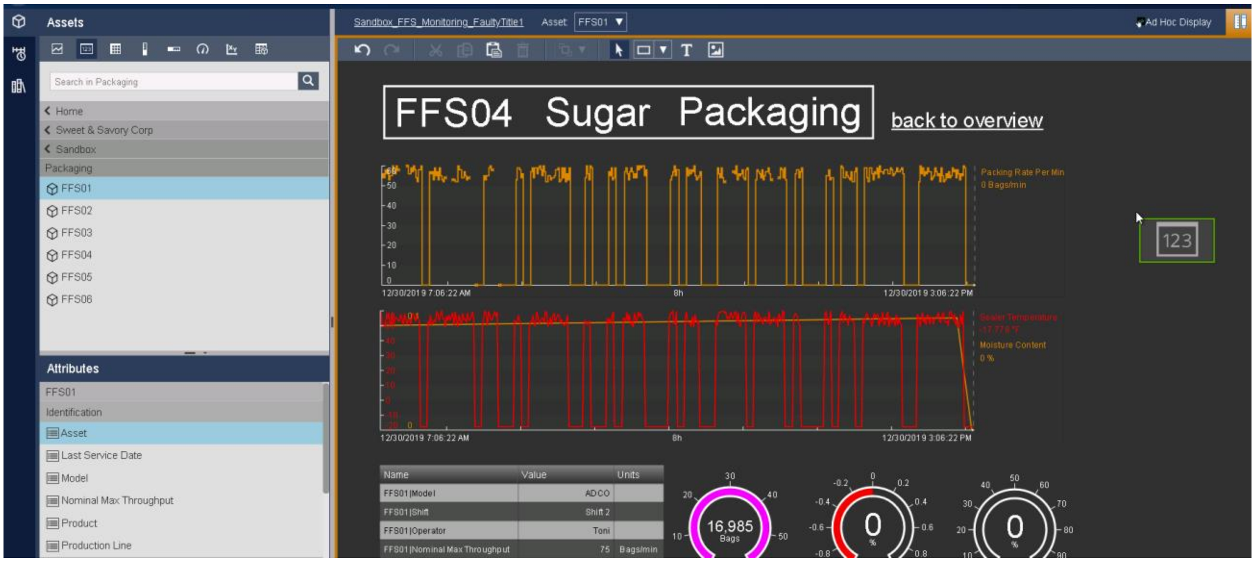
In order to make this title automatically update with the chosen asset, we need to use the AF attributes **Asset** and **Production Line**. These attributes are configured using the String Builder data reference, and point to the name of the asset itself and its parent asset. Let's first create a dynamic title and then we'll look behind the scenes at how String Builder attributes can retrieve element's names.

Step 1. Switch to Edit mode (button in the top-right corner of the display) and delete the existing title. You can simply select the title and hit the delete key.

Step 2. Open the asset pane on the left side of the display .

Step 3. Navigate to **Sandbox** → **Packaging** → **FFS01**.

Step 4. From the symbol pane, select the Value symbol in the top-left corner 123, then drag and drop the attribute **Asset** on the display (you should see this symbol 123 as in Figure 16).



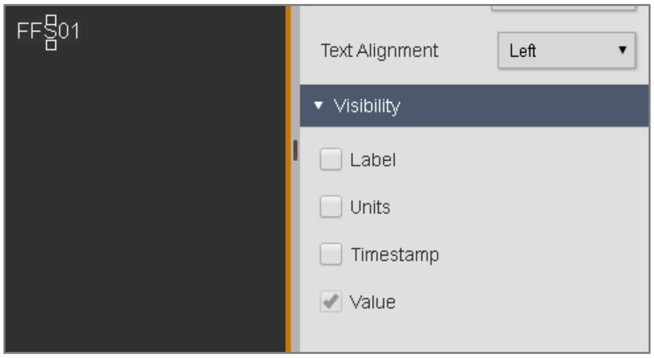
Name	Value	Units
FFS01|Model	ADCO	
FFS01|Shift	Shift 2	
FFS01|Operator	Toni	
FFS01|Nominal Max Throughput	75	Bags/min

Figure 16. Creating a dynamic title in PI Vision

The symbol will include the name of the attribute, its current value and the timestamp. In our case, we only need the value field.

Step 5. Right click on the symbol and select **Format Value**.

Step 6. Expand the **Visibility** tab and uncheck the Label, the Units and the Timestamp checkboxes (**leave just the Value** checkbox ticked as in Figure 17).



Text Alignment: Left

▾ Visibility

☐ Label

☐ Units

☐ Timestamp

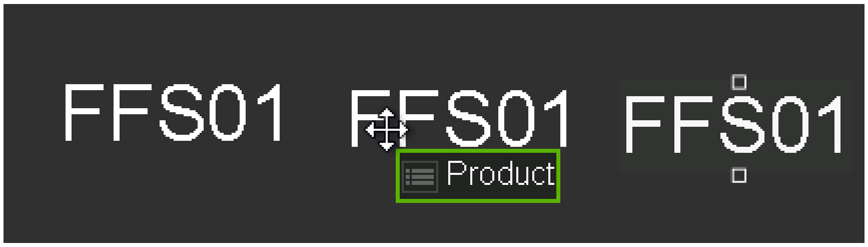
☑ Value

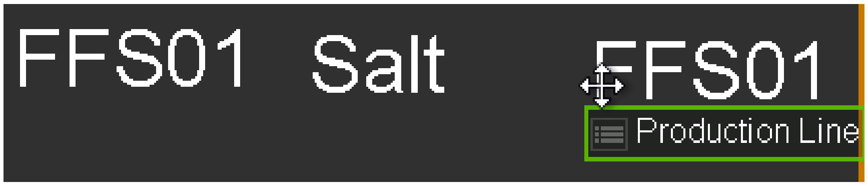
Figure 17. Visibility tab of the Value symbol

Step 7. Resize the symbol at will and place it at the top of the display as a title.

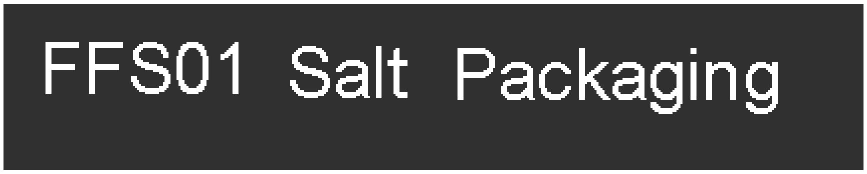
Step 8. Copy and paste the text box twice

Step 9. Drag the Product and the Production Line attributes on top of the copied symbols to replace their value (make sure you drag them on top of the existing symbols to replace their content):





Your new title should look as follows:

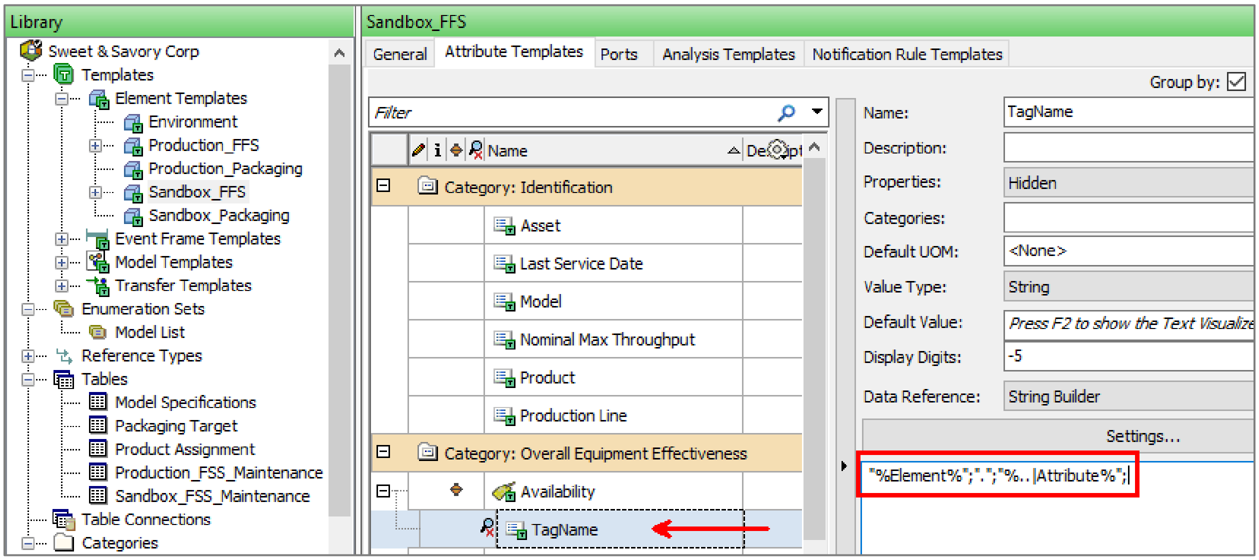


Step 10. Switch between assets using the Asset Dropdown Menu and verify that the title changes dynamically.

More on String Builder – A glimpse into substitution parameters

Let’s look at another example of how to use String Builder attributes. Remember the TagName attribute from Section 2.4.1? Move to the **Production_FFS** template and look at how these TagName attributes are built:

"%Element%";".";"%..|Attribute%";



Group by: ☑

Name: TagName

Description:

Properties: Hidden

Categories:

Default UOM: <None>

Value Type: String

Default Value: *Press F2 to show the Text Visualize*

Display Digits: -5

Data Reference: String Builder

"%Element%";".";"%..|Attribute%";

This syntax is used to concatenate:

(1) the name of the AF element that the attribute resides on (%Element%)

(2) the dot character (“.”)

(3) the name of the parent attribute with respect to the TagName attribute (%..|Attribute%)

Steps (1) and (3) use so-called **substitution parameters** (%Element% and %..|Attribute%), which minimize the need for manual configurations. String Builder can leverage substitution parameters, as well as attribute referencing and functions. If you want to learn more, we suggest you visit:

- String Builder data references
- Substitution parameters in data references

✓ Best Practice	String Builder is a powerful data reference type that helps you manipulate string values. String Builder attributes can be used to create dynamic titles in PI Client tools such as PI Vision, as well as to build references to PI Points on the Data Archive.

2.4.7. OPTIONAL - Categories

Asset Framework allows you to organize objects into categories, which you can define yourself. Their purpose is to help you find objects more easily. Objects can belong to multiple categories.

Each object type has its own categories. You cannot apply categories from one object type to an object of another type (e.g. you cannot apply an element category to a table). AF supports the following category types:

- Analysis
- Attribute
- Element
- Notification Rule
- Reference Type
- Table

When you search for an object in PI System Explorer, you can use the category as a filter to reduce the list of results. Categories are useful in many other situations, for example when:

- Selecting attributes in PI Vision
- configuring a Rollup analysis to pick up all attributes belonging to a given category
- filtering the analyses that you want to backfill / recalculate
- filtering AF objects when using the PI Integrator for Business Analytics or PI OLEDB Enterprise
- organizing and maintaining your Asset Framework server

In our sandbox environment, the AF attributes have already been categorized into:

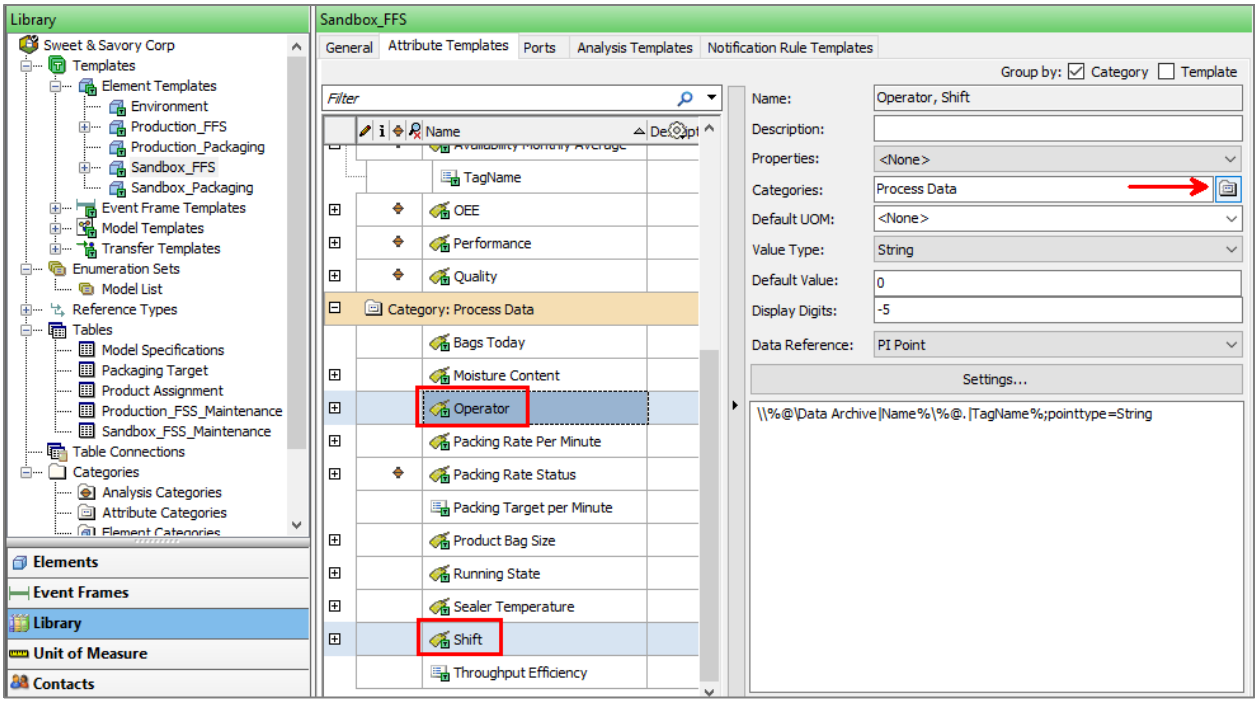
- **Identification**: static metadata providing contextual information about the asset
- **Overall Equipment Effectiveness**: attributes involved in the calculation of the OEE
- **Process Data**: all real-time data coming from the production line

The maintenance engineer noticed that a couple of attributes under the Process Data category could be better classified. He wants to create a dedicated category for the **Shift** and the **Operator** attributes and call this category **Shift Information**. Let's help him out!

Step 1. From the Library tab, select template Sandbox_FFS → Attribute Templates tab.

Step 2. Select both the Operator and the Shift attributes holding down the *Ctrl* keystroke.

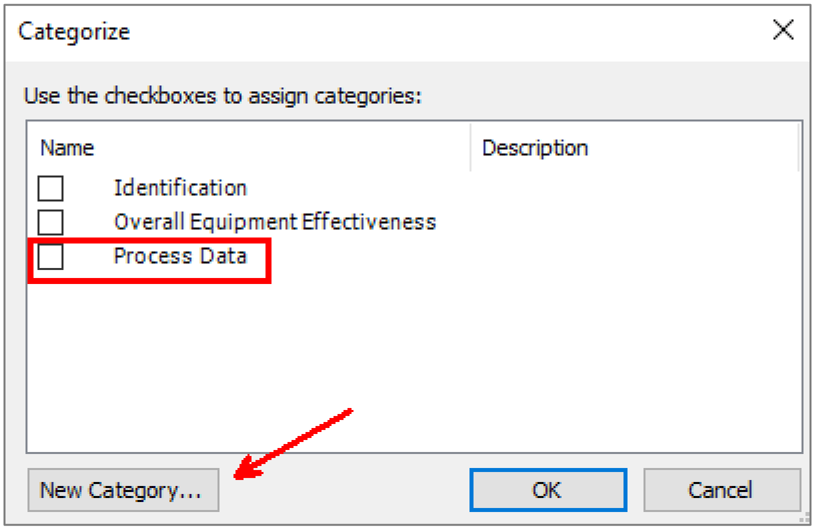
Step 3. Then click on the symbol on the right of the Categories field:



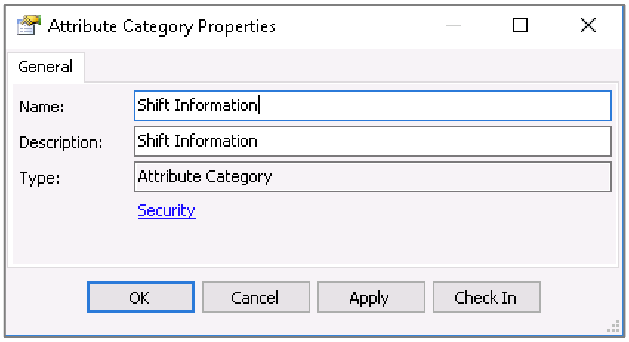
Name: Operator, Shift
Description:
Properties: <None>
Categories: Process Data
Default UOM: <None>
Value Type: String
Default Value: 0
Display Digits: -5
Data Reference: PI Point

\\%@\Data Archive|Name%\%@.|TagName%;pointtype=String

Step 4. Deselect Process Data and click on **New Category…** :



Step 5. Name the Category *Shift Information* and hit Check In and OK.


Attribute Category Properties

General

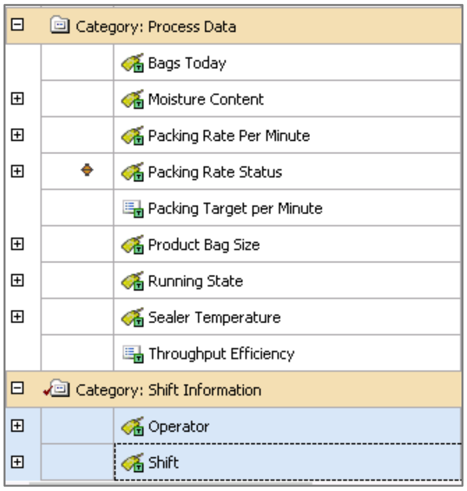
Name: Shift Information

Description: Shift Information

Type: Attribute Category

Step 6. Click OK again to close the *Categorize* pop up window.

You should see that the two chosen attributes now appear under the **Shift Information** category:

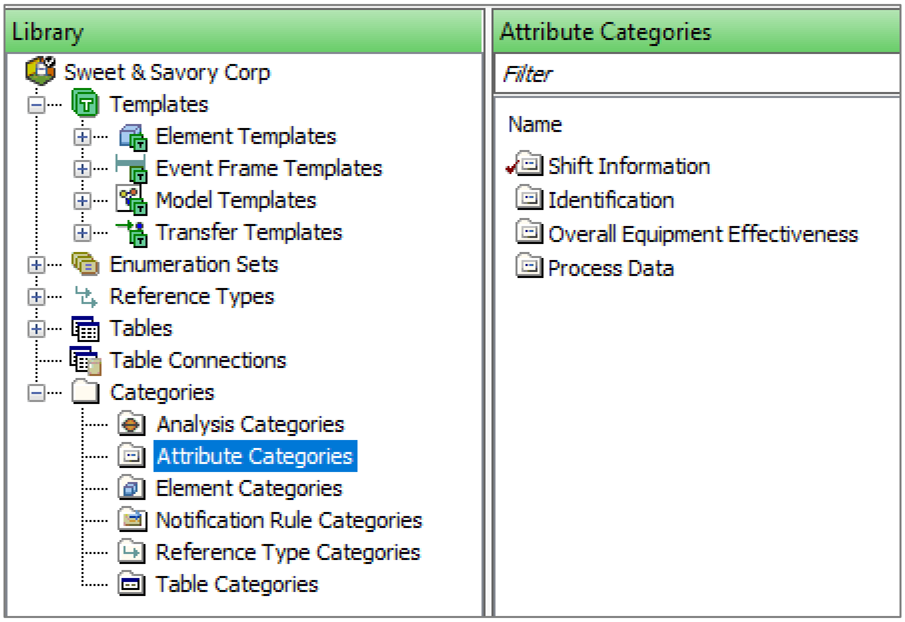

Category: Process Data

- Bags Today
- Moisture Content
- Packing Rate Per Minute
- Packing Rate Status
- Packing Target per Minute
- Product Bag Size
- Running State
- Sealer Temperature
- Throughput Efficiency

Category: Shift Information

- Operator
- Shift

You can explore the existing Categories in the Library by navigating to Categories → Attribute Categories. This can be useful when renaming or deleting Categories.



Library

- Sweet & Savory Corp
  - Templates
    - Element Templates
    - Event Frame Templates
    - Model Templates
    - Transfer Templates
  - Enumeration Sets
  - Reference Types
  - Tables
  - Table Connections
  - Categories
    - Analysis Categories
    - Attribute Categories
    - Element Categories
    - Notification Rule Categories
    - Reference Type Categories
    - Table Categories

Attribute Categories

Filter

Name
Shift Information
Identification
Overall Equipment Effectiveness
Process Data

Step 7. Important: Click on Check In **to save your changes.**

Step 8. Move to PI Vision, refresh the page and switch to Edit mode (button in the top-right corner).

Step 9. Navigate to one of the FFS units under the Sandbox environment and verify that the new categorization has been applied. Figure 18 shows that Shift and Operator now appear under the Shift Information category.

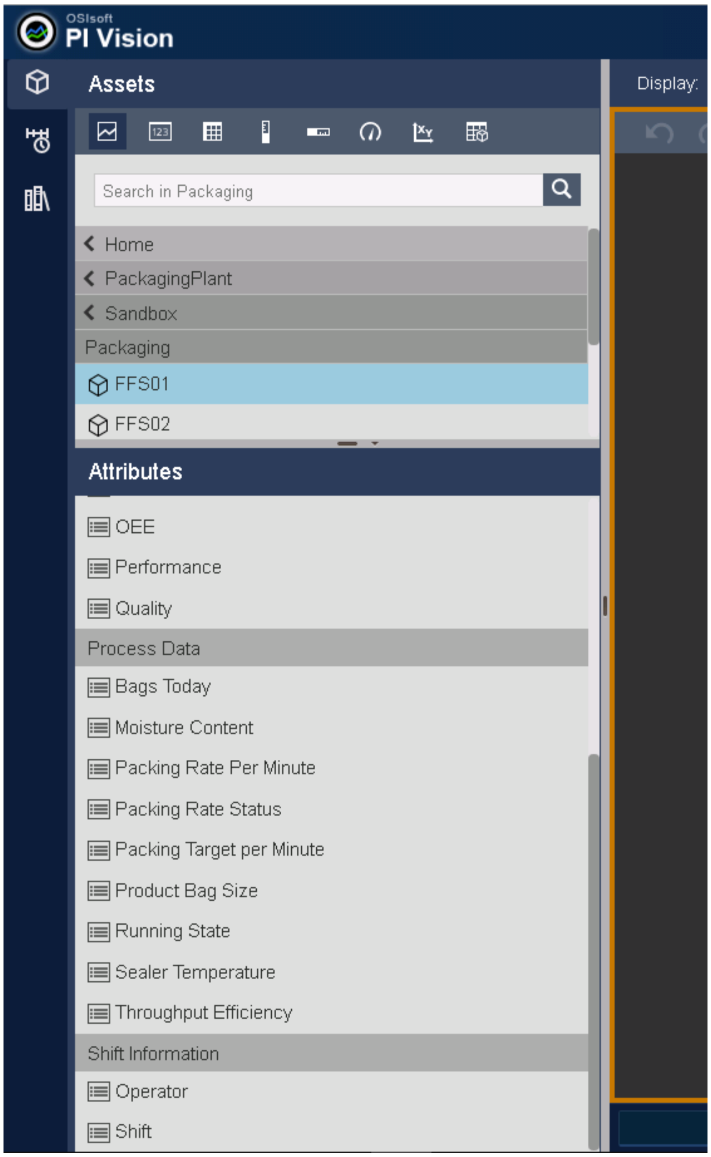


Figure 18. PI Vision displays AF attributes by category

✓ Best Practice	Use Categories to organize, manage and filter AF objects such as attributes, analyses and elements. This will help you search for what you need when building reports or displays with any PI Client tool such as PI Vision, PI DataLink and PI ProcessBook.

3. Working with Asset Analytics

This Chapter focuses on some important aspects of Asset Analytics. Section 3.1 introduces what Asset Analytics can do for your business, while Section 3.2 guides you through various scenarios where analyses can be improved in terms of performance and manageability.

3.1. Why Asset Analytics

What is the overall efficiency of a process? Which facility had the highest average production last week? And what about last month?

Users often run calculations on raw data to make key decisions such as which assets require maintenance, or to determine the profitability of a production site. However, these calculations often reside in spreadsheets, with data ranges, methods, or errors as numerous as the individuals who maintain them.

Asking one question can therefore lead to a range of different, and potentially contradictory, answers.

With Asset Analytics, you can easily configure server-side calculations with pre-built functions and *Intellisense* that will auto-populate suggestions as you type. Expressions can be basic, one-line calculations like the standard deviation of a voltage measurement, or multi-step calculations to determine overall equipment efficiency. Asset Analytics also provides a simple configuration option to write calculation results back to PI Points on either a timed (periodic) or events-driven schedule. All of this is possible with a no-coding-required user interface and enables large-scale deployment of calculations that are more manageable and powerful than could be handled within a typical spreadsheet.

For those using Performance Equations on the Data Archive, the same functionalities (and more!) are available in Asset Analytics where analyses can be easily deployed, backfilled, and managed at scale from a single user interface. (2)

Assets Analytics has become the calculation engine of choice for the PI System thanks to its ease of use and vast capabilities. With Assets Analytics, you can take advantage of your AF hierarchy and perform calculations on all your assets.

However, an overloaded **PI Analysis Service** (the Windows service running Asset Analytics) will result in lagging calculations, skipped evaluations, and slow backfilling/recalculations. This chapter provides best practices that will help you achieve optimum scalability and performance. For more information, please visit page: PI Analysis Service Best Practices (5), which also provides considerations on the PI Analysis

Service global configuration parameters and the Data Archive tuning parameters relevant to the PI Analysis Service.

3.2. Asset Analytics Best Practices - Directed activities

This Section presents some best practices applied to our case study. This time we don't start from PI Vision, as some of the potential issues we'll tackle might not have immediate repercussions on the end-user experience. However, we provide you other ways of assessing the impact of your configuration choices.

3.2.1. Use of Expression Variables

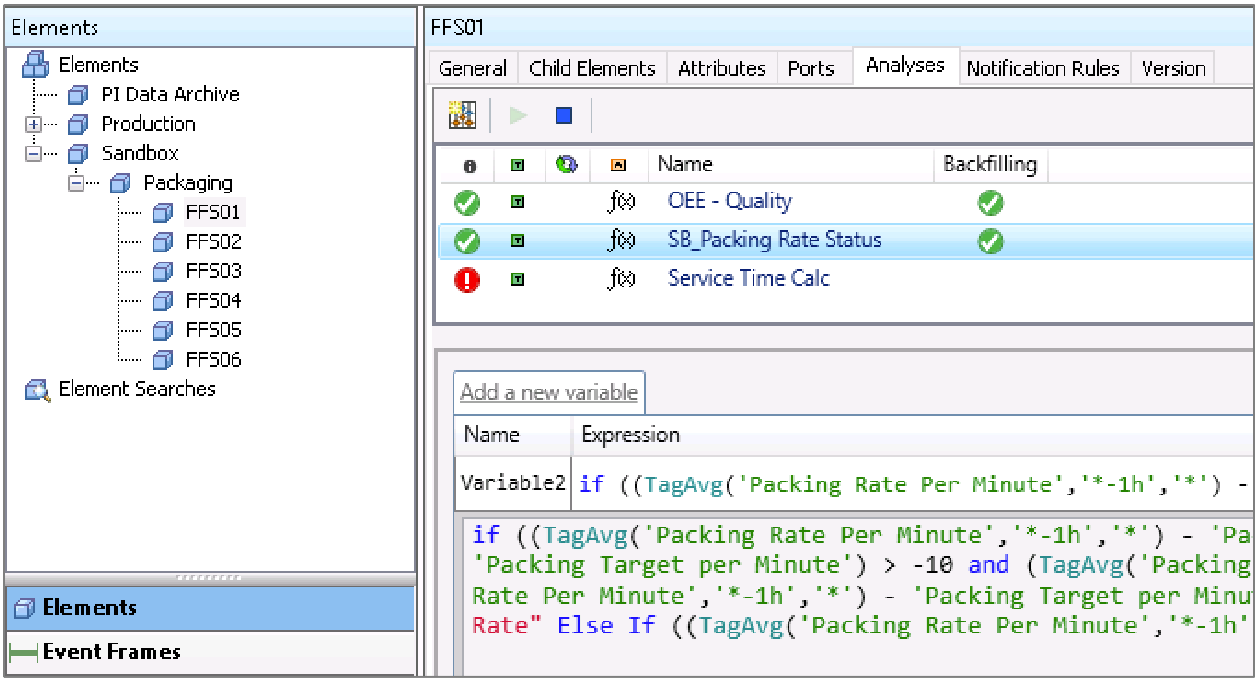
The chief process engineer wants to see in a glance whether the **hourly average** of the **Packing Rate per Minute** is meeting the target **Packing Target per Minute**.

The analysis should output one of the following messages:

- Far Behind Target Rate
- Behind Target Rate
- On Target
- Ahead of Target Rate
- Far Ahead of Target Rate

For this purpose, the analysis **SB_PackingRateStatus** has already been configured in the Sandbox environment.

Go to Sandbox → Packaging → FFS01, select the Analyses tab and check the configuration of the **SB_PackingRateStatus** analysis. You can click on the Expression field to expand it and view the analysis logic:



Name	Backfilling
OEE - Quality	✓
SB_Packing Rate Status	✓
Service Time Calc	

Name	Expression
Variable2	if ((TagAvg('Packing Rate Per Minute','*-1h','*') -

```
if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Pa
'Packing Target per Minute') > -10 and (TagAvg('Packing
Rate Per Minute','*-1h','*') - 'Packing Target per Minu
Rate" Else If ((TagAvg('Packing Rate Per Minute','*-1h'
```

The full logic is reported below:

```
if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per
Minute') < -10) then " Far Behind Target Rate" Else if ((TagAvg('Packing
Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > -10 and
(TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per
Minute') < 0) then "Behind Target Rate" else if ((TagAvg('Packing Rate
Per Minute','*-1h','*') - 'Packing Target per Minute') > 0 and
(TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per
Minute') < 10) then "Ahead of Target Rate" Else If ((TagAvg('Packing
Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > 10) then
"Far Ahead of Target Rate" Else "On Target"
```

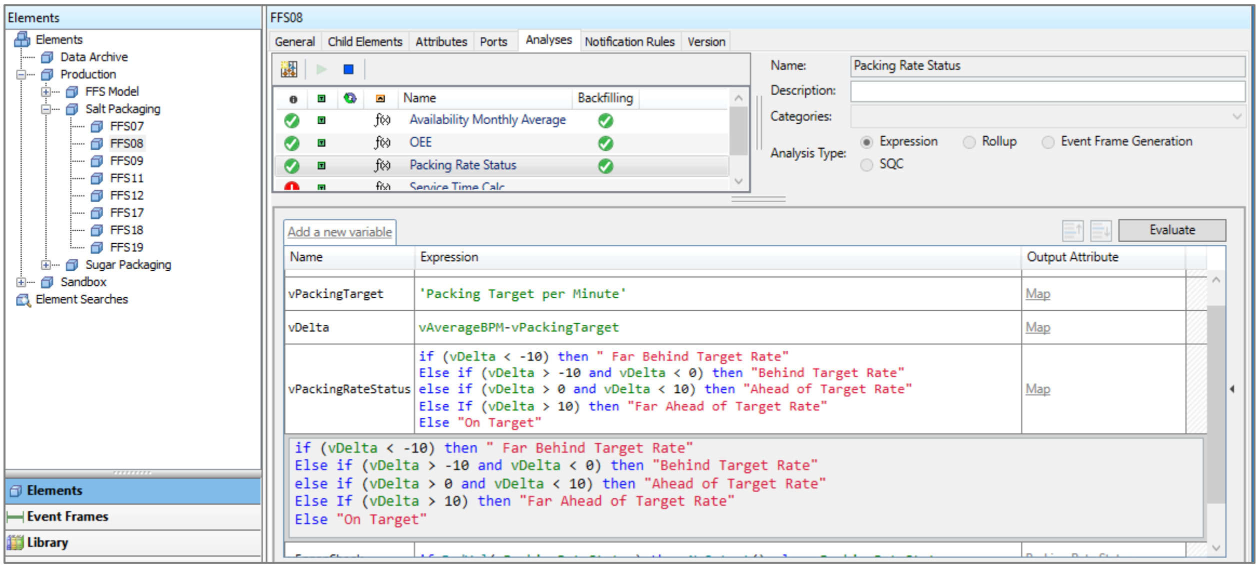
The expression looks rather lengthy and complex, but in fact it can be reduced to blocks of four steps:

1) calculate the hourly time-weighted average of the **Packing Rate per Minute** using the **TagAvg()** function
2) perform a subtraction between the hourly average and the **Packing Target per Minute**
3) perform a logic operation (< or >)
4) if the condition at point 3 evaluates to True, write a String value to the PI Point data reference attribute Packing Rate Status

Imagine that the hourly average of **Packing Rate per Minute** falls within the last condition of the IF statement ("*on target*"). How many times would we need to perform **steps 1 to 3** before deciding whether to write the result or to proceed to the next condition in the IF statement? [*4 times!*]

A much better option is to use **expression variables** to perform the calculation.

Look at the same analysis in one of the Elements under the **Production environment** (for example **FFS07**). The result of the calculation TagAvg('**Packing Rate per Minute**','*-1h', '*') is assigned to the variable **vAverageBPM**. In practice, the result is cached for the time needed to perform the evaluation. The analysis will only need to evaluate the hourly average once, and then perform steps 3 and 4 above. For more details on how the data cache of the PI Analysis Service works, please refer to The Analysis Service Data Cache (7).



Name	Backfilling
Availability Monthly Average	✓
OEE	✓
Packing Rate Status	✓
Service Time Calc	

Name: Packing Rate Status
Description:
Categories:
Analysis Type: ● Expression ○ Rollup ○ Event Frame Generation ○ SQC

Name	Expression	Output Attribute
vPackingTarget	'Packing Target per Minute'	Map
vDelta	vAverageBPM-vPackingTarget	Map
vPackingRateStatus	if (vDelta < -10) then " Far Behind Target Rate" Else if (vDelta > -10 and vDelta < 0) then "Behind Target Rate" else if (vDelta > 0 and vDelta < 10) then "Ahead of Target Rate" Else If (vDelta > 10) then "Far Ahead of Target Rate" Else "On Target"	Map

```
if (vDelta < -10) then " Far Behind Target Rate"
Else if (vDelta > -10 and vDelta < 0) then "Behind Target Rate"
else if (vDelta > 0 and vDelta < 10) then "Ahead of Target Rate"
Else If (vDelta > 10) then "Far Ahead of Target Rate"
Else "On Target"
```

Tip

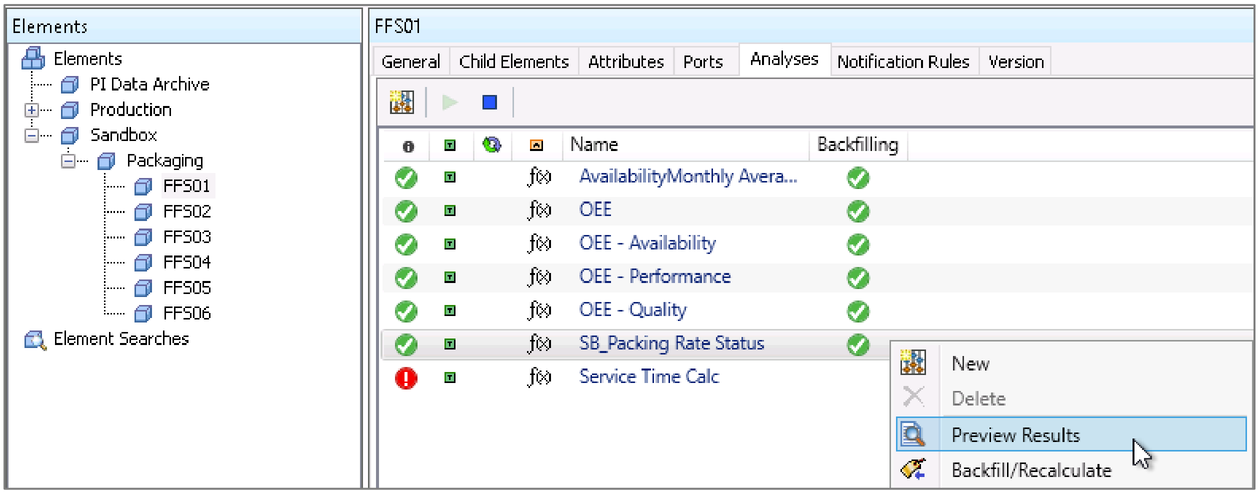
Use the little "v" in front of your variables, so you can easily distinguish expression variables versus AF attributes. Also, notice that when you recall a variable name you should not use single quotes.

You might be wondering… does it really make a difference in terms of performance of the PI Analysis Service?

In order to test whether this configuration choice makes any difference:

Step 1. Move to element **FFS01** → Analyses tab.

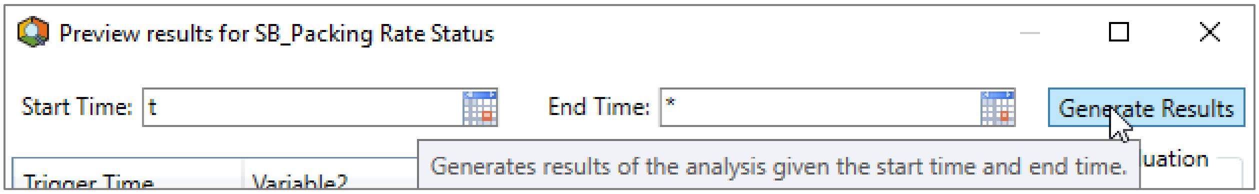
Step 2. Right-click on the analysis **SB_Packing Rate Status** and select **Preview Results.**



	Name	Backfilling
✓	AvailabilityMonthly Avera...	✓
✓	OEE	✓
✓	OEE - Availability	✓
✓	OEE - Performance	✓
✓	OEE - Quality	✓
✓	SB_Packing Rate Status	✓
!	Service Time Calc	

Step 3. Select the desired time range (e.g. Start Time = t, End Time = *)

Step 4. Click on **Generate Results** on the top right of the window.



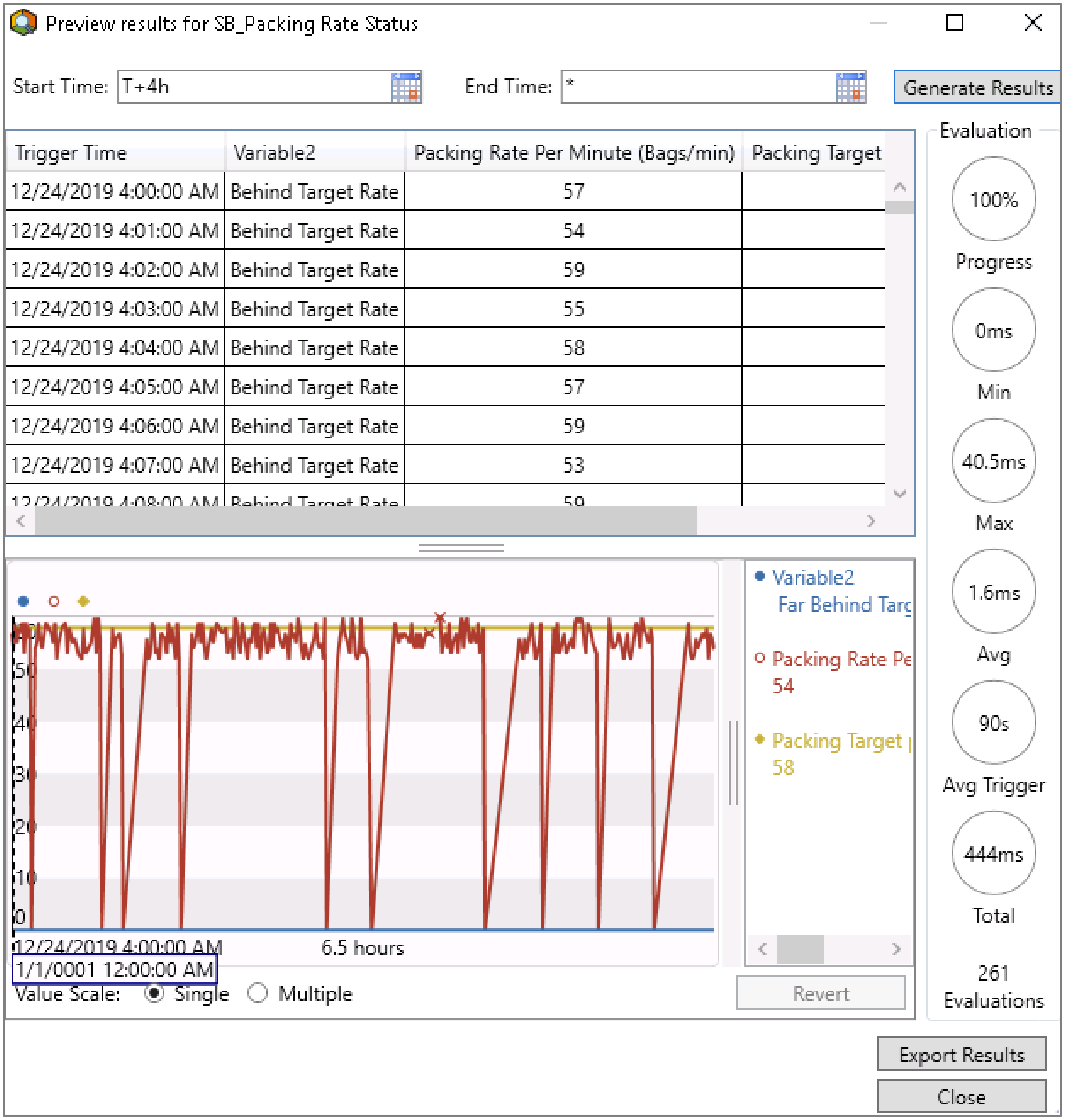
Start Time: t

End Time: *

Generates results of the analysis given the start time and end time.

The statistic that we are most interested in is the **Average** evaluation time.

In the example below, we see that the analysis in the Sandbox environment takes on average 1.6 milliseconds to be evaluated (Figure 19), while the analysis under the Production environment takes on average only 0.6 milliseconds (Figure 20). We reduce by over 62% the evaluation time just by using the expression variables!



Preview results for SB_Packing Rate Status

Start Time: T+4h End Time: * Generate Results

Trigger Time	Variable2	Packing Rate Per Minute (Bags/min)	Packing Target
12/24/2019 4:00:00 AM	Behind Target Rate	57	
12/24/2019 4:01:00 AM	Behind Target Rate	54	
12/24/2019 4:02:00 AM	Behind Target Rate	59	
12/24/2019 4:03:00 AM	Behind Target Rate	55	
12/24/2019 4:04:00 AM	Behind Target Rate	58	
12/24/2019 4:05:00 AM	Behind Target Rate	57	
12/24/2019 4:06:00 AM	Behind Target Rate	59	
12/24/2019 4:07:00 AM	Behind Target Rate	53	
12/24/2019 4:08:00 AM	Behind Target Rate	59	

Evaluation

100%
Progress

0ms
Min

40.5ms
Max

1.6ms
Avg

90s
Avg Trigger

444ms
Total

261
Evaluations

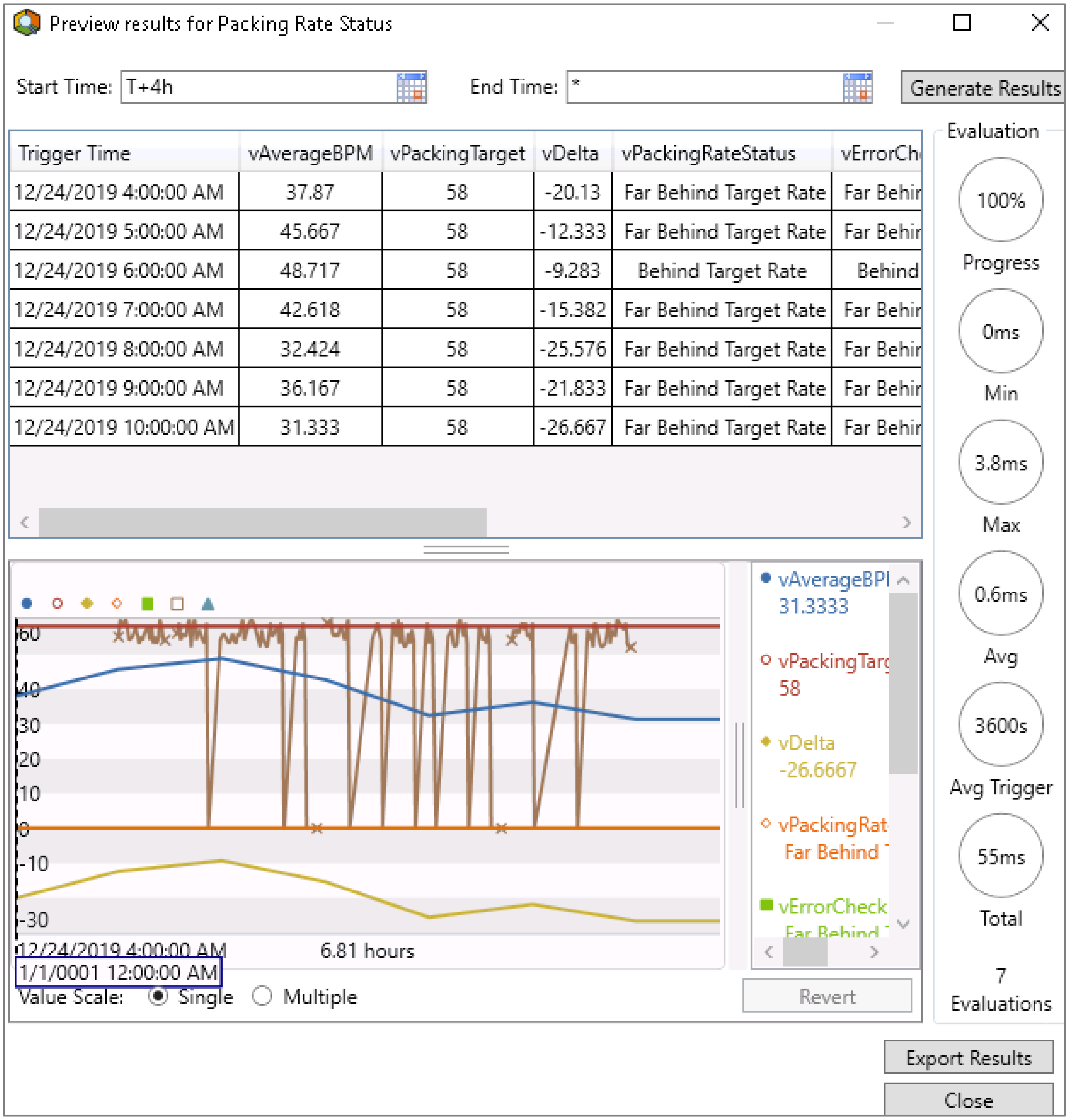
● Variable2
Far Behind Targ

○ Packing Rate Pe
54

◆ Packing Target |
58

Value Scale: ● Single ○ Multiple

Figure 19. Preview Results for the analysis “SB_Packaging Rate Status” in the Sandbox environment



Start Time: T+4h

End Time: *

Trigger Time	vAverageBPM	vPackingTarget	vDelta	vPackingRateStatus	vErrorCh
12/24/2019 4:00:00 AM	37.87	58	-20.13	Far Behind Target Rate	Far Behir
12/24/2019 5:00:00 AM	45.667	58	-12.333	Far Behind Target Rate	Far Behir
12/24/2019 6:00:00 AM	48.717	58	-9.283	Behind Target Rate	Behind
12/24/2019 7:00:00 AM	42.618	58	-15.382	Far Behind Target Rate	Far Behir
12/24/2019 8:00:00 AM	32.424	58	-25.576	Far Behind Target Rate	Far Behir
12/24/2019 9:00:00 AM	36.167	58	-21.833	Far Behind Target Rate	Far Behir
12/24/2019 10:00:00 AM	31.333	58	-26.667	Far Behind Target Rate	Far Behir

Evaluation

100% Progress

0ms Min

3.8ms Max

0.6ms Avg

3600s Avg Trigger

55ms Total

7 Evaluations

● vAverageBPI 31.3333

○ vPackingTarg 58

◆ vDelta -26.6667

◇ vPackingRat Far Behind

■ vErrorCheck Far Behind

12/24/2019 4:00:00 AM 6.81 hours

1/1/0001 12:00:00 AM

Value Scale: ● Single ○ Multiple

Figure 20. Preview Results for the analysis "Packaging Rate Status" in the Production environment

A similar example is available in the PI Square post Asset Analytics Best Practices - Part 1: Use Variables (4). Commenting on the example provided on the post, our AF & Analytics specialist states:

«The analysis with the repeating function took ~6 times longer to evaluate in my system. You may be telling yourself, even though it's slower, it's still only takes about 34ms to complete which isn't much and this isn't a problem.

However, what's extremely important here is to **think about scalability!** If the *[ill-configured]* analysis was based on an analysis template and there were hundreds or thousands of other analyses based on the same analysis template, the performance difference at the group level (all analyses based on the same template) would be **on the order of seconds, perhaps even minutes**. »

NERD ALERT - Additional consideration: client vs server evaluations

In our example we used the **Preview Results** tab, which gives us evaluation statistics for **client-side evaluations**. In other words, when you click on Preview Results you are **not** calling the PI Analysis Service itself; instead, you are using the **resources of the client machine** you are on. In our case, we are working in PI System Explorer on the PI Server machine itself, but normally you would open it from your client machine.

The evaluation statistics provided by the Preview Results window are often a bit *pessimistic*, as the machine hosting the PI Analysis Service is usually more performing (in terms of memory and processor speed) than the client machine you use. As a rule of thumb, we suggest making sure that the average (client-side) evaluation time is significantly below the triggering frequency (see next section 3.2.2 for more on analysis scheduling).

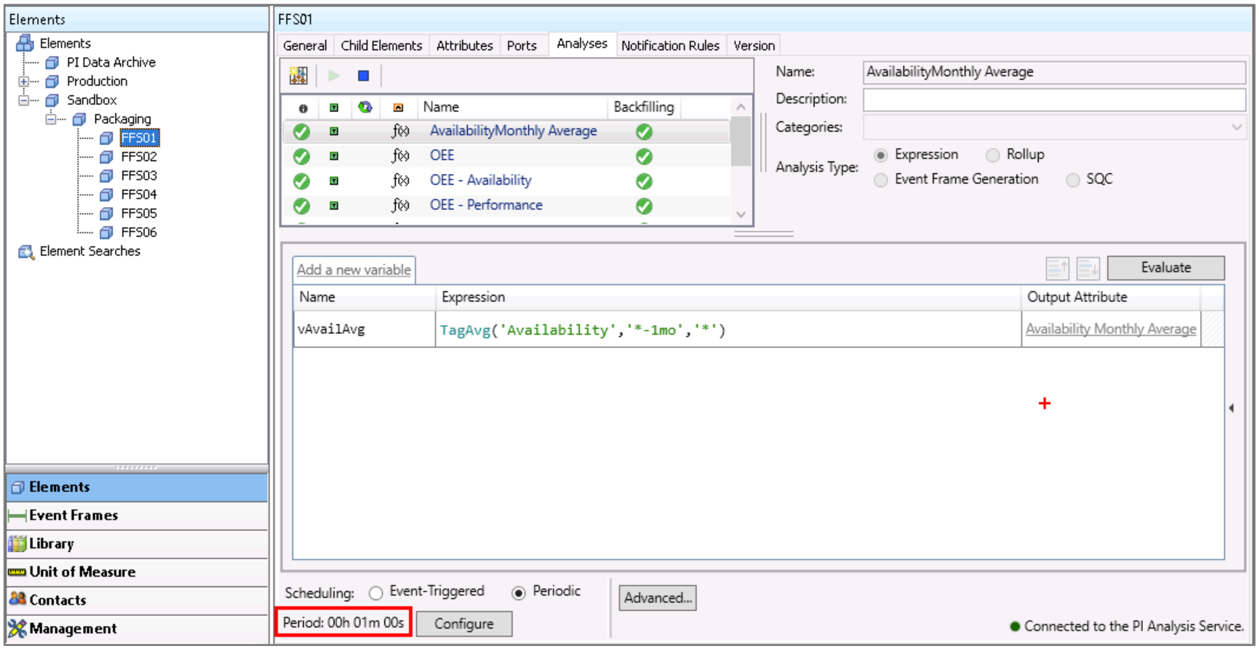
✓ Best Practice	Use expression variables whenever you want to repeat a block of your analysis logic, such as a summary operation for the same tag and time span. This will improve the performance of the PI Analysis Service as well as simplify the analysis management.

3.2.2. Analysis Scheduling

The chief maintenance engineer wants to keep track of the **Availability Monthly Average** for the purpose of identifying which FFS units have undergone the longest monthly downtime.

He needs the results of this calculation twice a week for his meetings with the maintenance team.

Let’s move to FFS01 and look at the analysis **Availability Monthly Average**. This analysis uses the function **TagAvg()** and spans over the last month of data. What’s important to notice here is the **Scheduling** configuration: the analysis is set to trigger on a periodic basis with a **1-minute frequency**!



Name	Backfilling
AvailabilityMonthly Average	✓
OEE	✓
OEE - Availability	✓
OEE - Performance	✓

Name: AvailabilityMonthly Average

Description:

Categories:

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

Name	Expression	Output Attribute
vAvailAvg	TagAvg('Availability','*-1mo','*')	Availability Monthly Average

Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 01m 00s

Connected to the PI Analysis Service.

Why should we trigger the analysis every minute if the results are needed only twice a week?

These 1-minute evaluations pose an unnecessary workload on the PI Analysis Service. The impact becomes even more significant when backfilling or recalculating the analysis over a long period of time, as the PI Analysis Service will have to perform many evaluations “in bulk”. For example, given a 1-minute frequency, backfilling over the last 6 months would yield 259 200 evaluations. If we set it to daily frequency, we go down to 180 evaluations for the same time span (less than 0.07% of the initial number of evaluations).

In order to change the scheduling of the analysis:

Step 1. Move to Library → Templates → Element Templates → **Sandbox_FFS.**

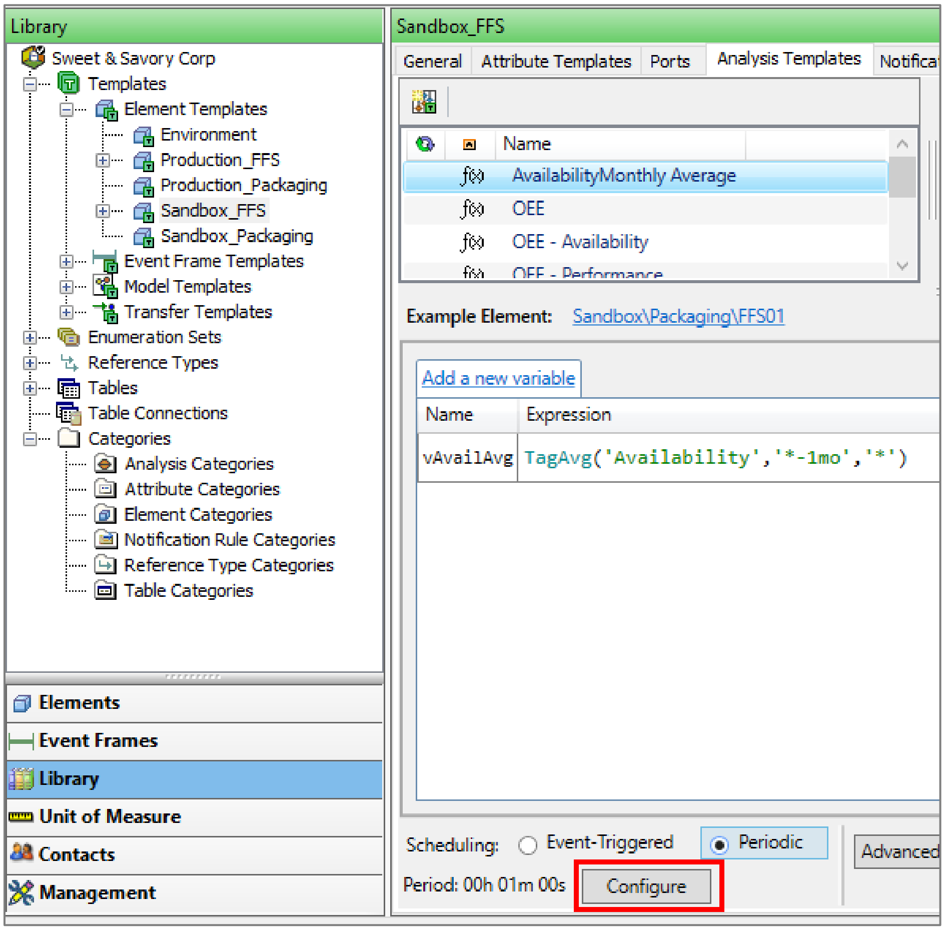
Step 2. Open the Analysis Templates tab.

Step 3. Click on the analysis named Availability Monthly Average (Figure 21).

Step 4. Click on the **Configure** button and choose **Daily** (Figure 22).

Step 5. Under the same tab, we can also specify the time of day we want to run the analysis at. Let's set it to **1 AM** (unless otherwise required, it is good practice to set offsets compared to the default time of 12 AM, so that not all daily calculations get triggered at the same time).

Step 6. Once you are done, click OK and Check In.



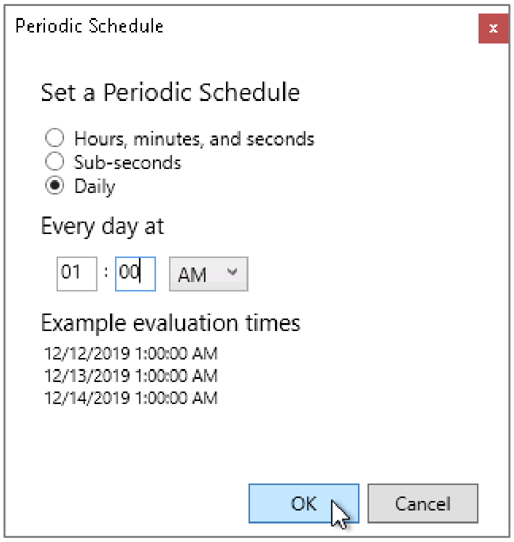
Example Element: Sandbox\Packaging\FFS01

Name	Expression
vAvailAvg	TagAvg('Availability','*-1mo','*')

Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 01m 00s

Figure 21. Availability Monthly Average configuration



Periodic Schedule

Set a Periodic Schedule

○ Hours, minutes, and seconds
○ Sub-seconds
● Daily

Every day at

01 : 00 AM

Example evaluation times

12/12/2019 1:00:00 AM
12/13/2019 1:00:00 AM
12/14/2019 1:00:00 AM

Figure 22. Periodic Schedule tab

If you would like to learn more about the impact of data density and scheduling on the PI Analysis Service, please visit these PI Square posts: Asset Analytics Best Practices - Part 2: Data Density and Data Pattern (9) and Asset Analytics Best Practices - Part 5: Scheduling (10).

NERD ALERT – Exit() function and data density

The analysis could have been configured to skip the evaluation if the current day is not the day when the results are needed. For example, the logic could be:

```
if weekday('*') <> 2 or weekday('*') <> 6 then Exit() else TagAvg(‘Availability’, ‘*-1mo’, ‘*’)
```

NB: The **Exit()** function has been introduced in release **2018 SP2** and is particularly useful when performing high-workload calculations such as summary calculations over extensive periods of time.

Finally, if the input data is very dense, you could split the average calculation in the following way:

- Calculate the daily average (and save the output result in a PI Point). This analysis could be scheduled as Event-Triggered.
- Calculate the monthly average using, as input, the output of the daily calculation. This analysis could be scheduled as Periodic and run daily.

This will reduce the total amount of data used for the monthly calculation by a factor 30!

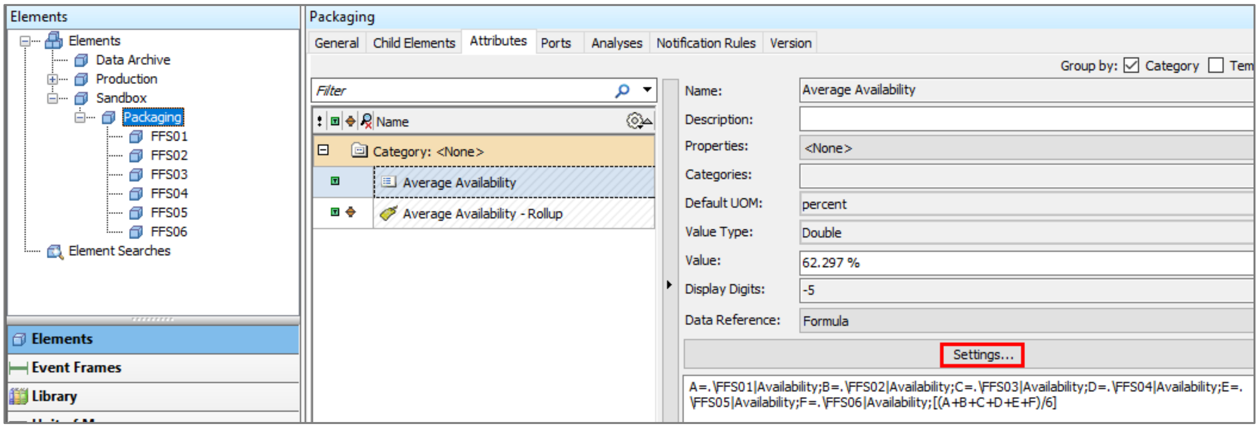
✓ Best Practice	Don't run an analysis more frequently than it needs to be run, as needless evaluations may pose a significant load on the PI Analysis Service.

3.2.3. Rollup Analyses

A rollup analysis calculates statistics such as sum and average of selected attributes associated with an element.

The chief process engineer needs to monitor the **Average Availability** of the FFS units at the Production Line level. One of his colleagues suggested that he uses a Formula data reference attribute, on the basis that it’s easy to set up and it evaluates only *on the fly*, when the result is requested. In order to create a Formula, he assigns each input to a parameter and then writes the equation as in Figure 23.

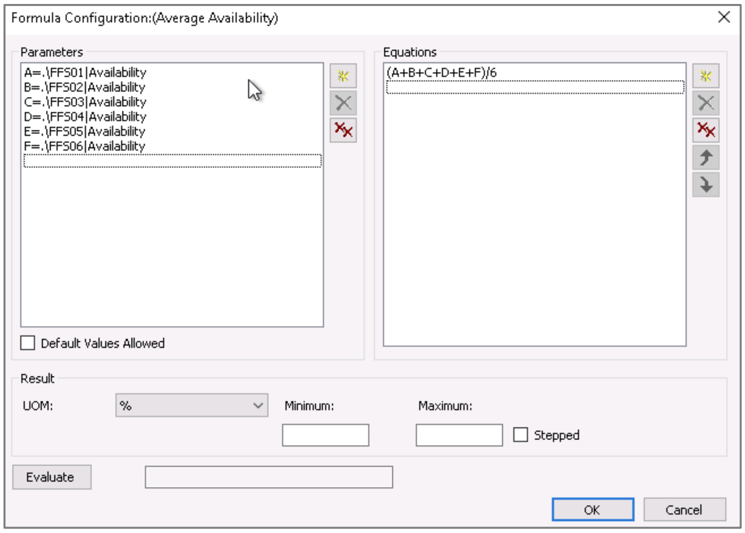
Click on the **Settings** button to display the parameters and equation in a more user-friendly way (Figure 24):



Name: Average Availability
Description:
Properties: <None>
Categories:
Default UOM: percent
Value Type: Double
Value: 62.297 %
Display Digits: -5
Data Reference: Formula

```
A=.\FFS01|Availability;B=.\FFS02|Availability;C=.\FFS03|Availability;D=.\FFS04|Availability;E=.
\FFS05|Availability;F=.\FFS06|Availability;[(A+B+C+D+E+F)/6]
```

Figure 23. Average Availability calculation – Formula data reference



Formula Configuration:(Average Availability)

Parameters

A=.\FFS01|Availability
B=.\FFS02|Availability
C=.\FFS03|Availability
D=.\FFS04|Availability
E=.\FFS05|Availability
F=.\FFS06|Availability

□ Default Values Allowed

Equations

(A+B+C+D+E+F)/6

Result

UOM: %

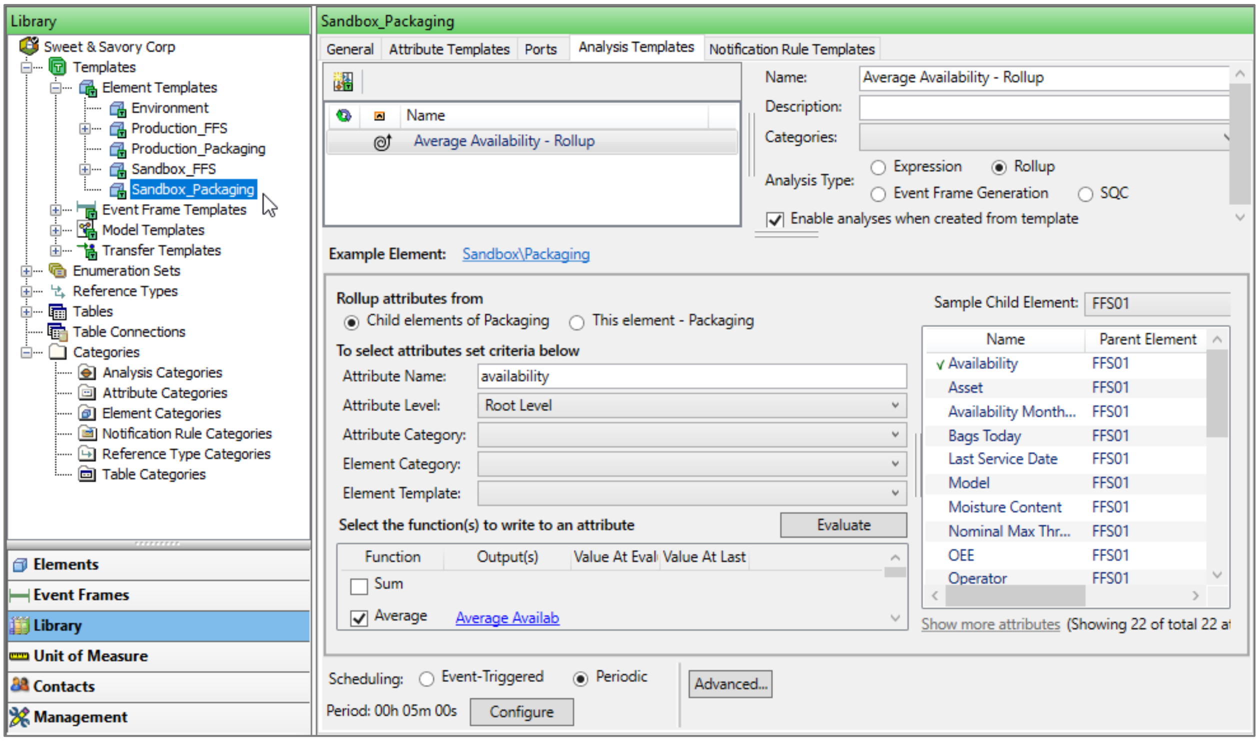
Minimum:

Maximum:

□ Stepped

Figure 24. Average Availability calculation – Formula settings tab

After implementing the Formula, our chief production engineer finds out an alternative way of configuring this analysis, i.e. using a Rollup analysis within Asset Analytics (Figure 25). The element Packaging uses the AF element template **Sandbox_Packaging**.



Sandbox_Packaging

Name
Average Availability - Rollup

Name: Average Availability - Rollup

Description:

Categories:

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

☑ Enable analyses when created from template

Example Element: Sandbox\Packaging

Rollup attributes from
● Child elements of Packaging ○ This element - Packaging

To select attributes set criteria below

Attribute Name: availability

Attribute Level: Root Level

Attribute Category:

Element Category:

Element Template:

Select the function(s) to write to an attribute

Function	Output(s)	Value At Eval	Value At Last
□ Sum			
☑ Average	Average Availab		

Sample Child Element: FFS01

Name	Parent Element
✓ Availability	FFS01
Asset	FFS01
Availability Month...	FFS01
Bags Today	FFS01
Last Service Date	FFS01
Model	FFS01
Moisture Content	FFS01
Nominal Max Thr...	FFS01
OEE	FFS01
Operator	FFS01

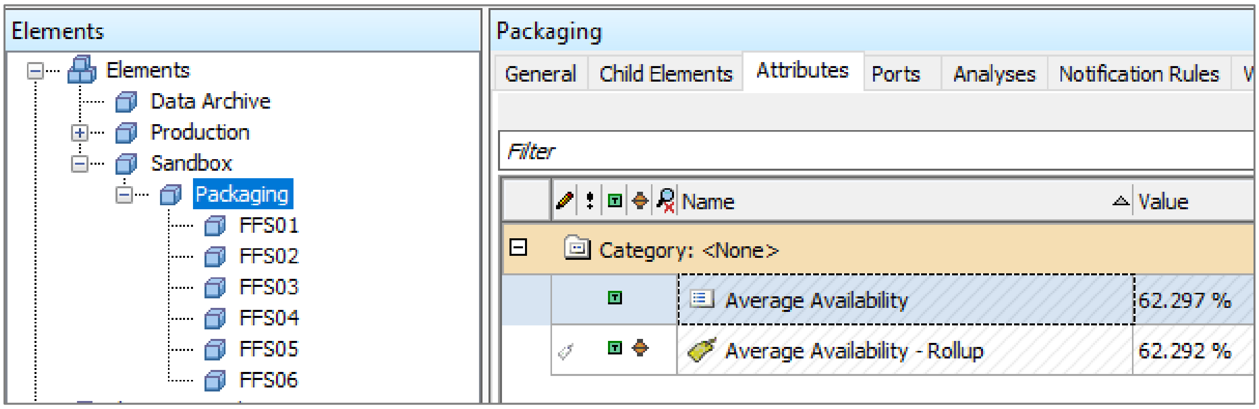
Show more attributes (Showing 22 of total 22 a

Scheduling: ○ Event-Triggered ● Periodic

Period: 00h 05m 00s

Figure 25. Average Availability – Rollup analysis

The chief production engineer tests both analyses in the Sandbox environment and it looks like they provide the same results down to the first decimal:



Name	Value
Average Availability	62.297 %
Average Availability - Rollup	62.292 %

Figure 26. Average availability – comparison between Formula data reference and Rollup analysis

Which of the two approaches would you suggest using and why?

See what happens when you delete a couple of Elements from the Sandbox environment.

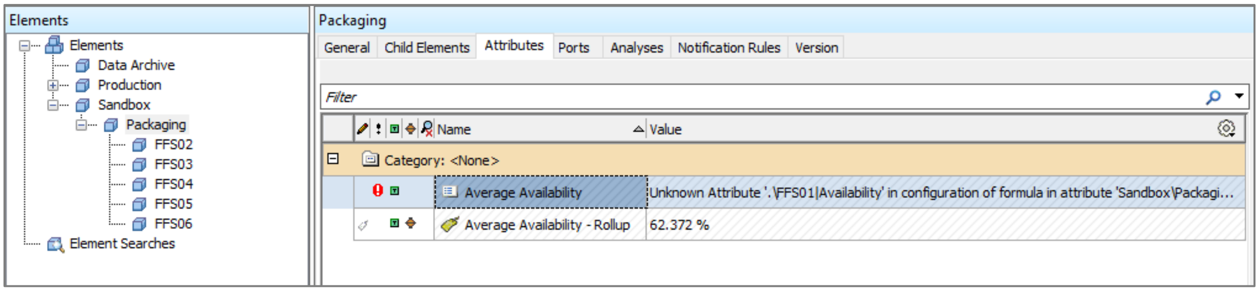
1. Move to the Elements tab and select **FFS01**.
2. Right-click and select **Delete…** In the opening pop-up windows choose:

```
Permanently delete; this action is irreversible
```

3. Now hit Check In and Refresh.

You will notice that the rollup analysis that uses a Formula data reference shows error message:

```
Unknown Attribute '.\FFS01|Availability' in configuration of formula
in attribute 'Sandbox\Packaging|Average Availability'.
```



Name	Value
Average Availability	Unknown Attribute '.\FFS01|Availability' in configuration of formula in attribute 'Sandbox\Packagi…
Average Availability - Rollup	62.372 %

Figure 27. Error message following the deletion of FFS01.

The error message pops us because the Formula data reference is not “smart” and does not update whenever a Child Element of the Packaging element is added or removed. Moreover, the “FFS count” at the denominator of the Formula expression (6, i.e. the number of FFS units) must be manually modified if the unit count changes.

On the contrary, a *proper* Rollup analysis built with Asset Analytics identifies its input attributes each time it is executed, and automatically includes any new attributes that meet its selection criteria (or removes inputs that are no longer available).

If you haven’t had a chance to work with Rollup analyses, take a few minutes to examine the configuration of the analysis **Average Availability – Rollup**.

You will notice that you can apply the rollup to Child Elements or to attributes of the Element itself.

You can also apply filters such as categories and element templates (for example, you could create a Rollup analysis at the Production level that considers only the Elements built on the template **Sugar Production_FFS**).

Which scenarios in your industry can you think of where rollup analyses could be used?

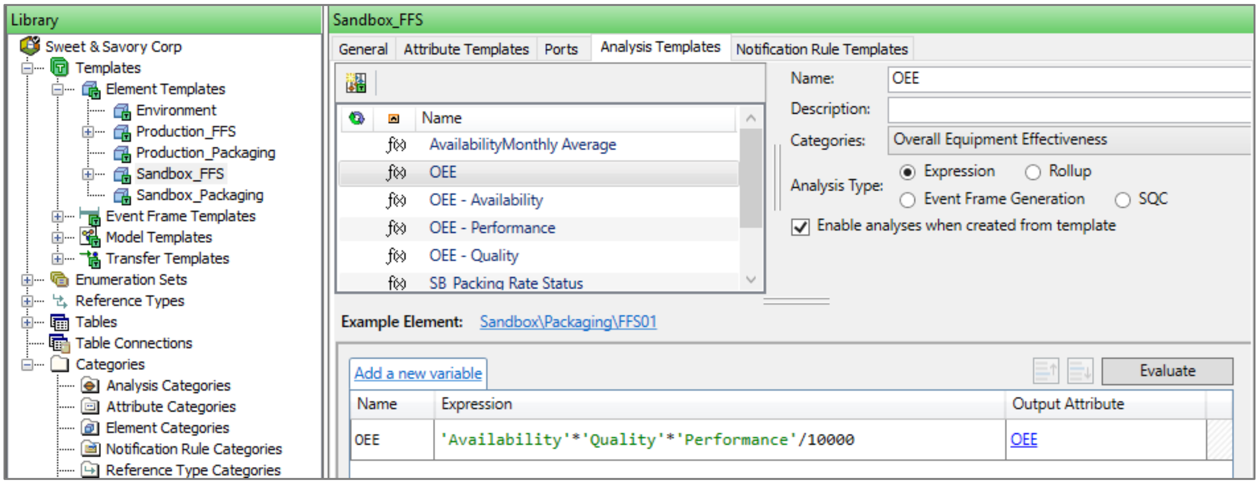
✓ Best Practice	You can add or delete elements or attributes in your hierarchy without the need to update Rollup analyses. Because a Rollup identifies input attributes each time it is executed, it automatically includes any new attributes that meet its selection criteria.

The following two sections are marked as *nerd alert*. Feel free to go through them should you still have time, or save them for when you are back home with your PI World gadgets!

3.2.4. NERD ALERT – Analysis Organization and Consolidation

Organizing analyses in a clear manner is not just about the aesthetics.

Take for example the OEE analysis configured in template **Sandbox_FFS** and in template **Production_FFS**.



AvailabilityMonthly Average
OEE
OEE - Availability
OEE - Performance
OEE - Quality
SB Packing Rate Status

Name: OEE

Description:

Categories: Overall Equipment Effectiveness

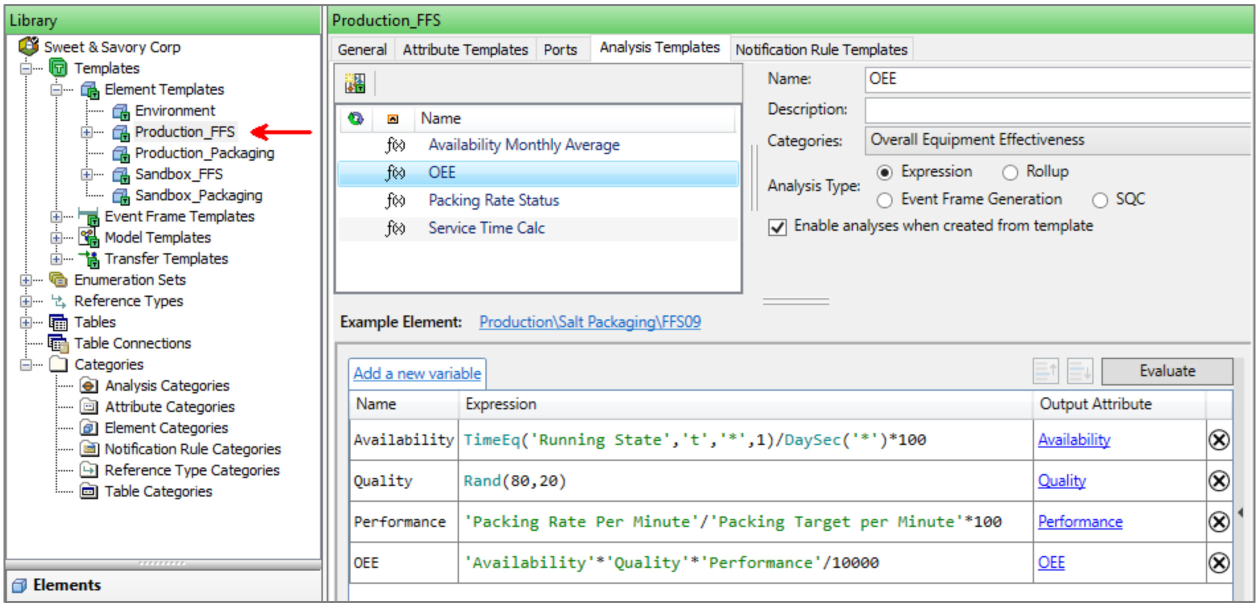
Analysis Type: ● Expression ○ Rollup ○ Event Frame Generation ○ SQC

☑ Enable analyses when created from template

Example Element: Sandbox\Packaging\FFS01

Name	Expression	Output Attribute
OEE	'Availability'*'Quality'*'Performance'/10000	OEE

Figure 28. Sandbox_FFS



Availability Monthly Average
OEE
Packing Rate Status
Service Time Calc

Name: OEE

Description:

Categories: Overall Equipment Effectiveness

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

☑ Enable analyses when created from template

Example Element: Production\Salt Packaging\FFS09

Name	Expression	Output Attribute
Availability	TimeEq('Running State','t','*',1)/DaySec('*')*100	Availability
Quality	Rand(80,20)	Quality
Performance	'Packing Rate Per Minute'/'Packing Target per Minute'*100	Performance
OEE	'Availability'*'Quality'*'Performance'/10000	OEE

Figure 29. Production_FFS

The OEE calculation in the Sandbox environment is split up into four distinct analyses, while in the Production environment there is only one analysis comprised of four variables.

You can choose between consolidating and splitting analysis expressions by considering the three criteria below:

- **Dependency**. If one analysis output depends on the results of another analysis, it often makes more sense to consolidate the analyses. This ensures that the dependent expression gets evaluated only when all the inputs have been evaluated using the most recent data. In our example, the OEE should only be calculated when Availability, Quality and Performance have been calculated.

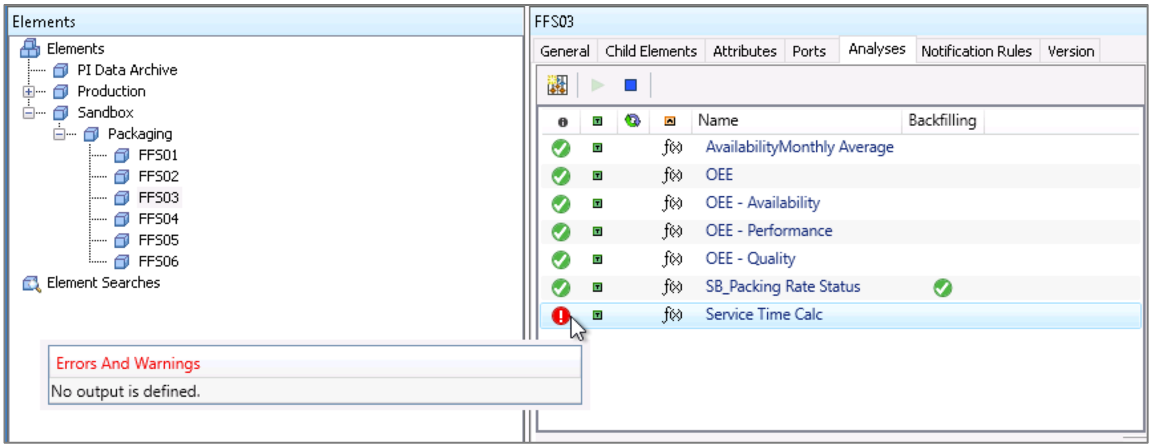
- **Scheduling**. If a group of related expressions should be evaluated at the same time, it is usually best to consolidate them.
  However, if some of these expressions need to be evaluated at a much higher frequency, it is better to create independent analyses to avoid needless evaluations of the slow-paced expressions.
- **Manageability**. Consolidating analyses facilitates management and troubleshooting. In case the analysis is failing, using the **Evaluate** or **Preview Results** options will guide you to the faulty expression.

✓ Best Practice	When deciding whether to consolidate several expressions under the same analysis, take into consideration Dependency, Scheduling and Manageability.

3.2.5. NERD ALERT - Analyses in Error

By now you'll have noticed that the **Service Time Calc** analyses are all in error state (Figure 30).

The chief maintenance engineer had planned to use these analyses to calculate the number of days since the last maintenance service. However, other tasks took priority and the analyses were left in a half-cooked state.



Elements
- Elements
  - PI Data Archive
  - Production
  - Sandbox
    - Packaging
      - FFS01
      - FFS02
      - FFS03
      - FFS04
      - FFS05
      - FFS06
- Element Searches

FFS03

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

				Name	Backfilling
✓	▪		f(x)	AvailabilityMonthly Average	
✓	▪		f(x)	OEE	
✓	▪		f(x)	OEE - Availability	
✓	▪		f(x)	OEE - Performance	
✓	▪		f(x)	OEE - Quality	
✓	▪		f(x)	SB_Packing Rate Status	✓
!	▪		f(x)	Service Time Calc	

Errors And Warnings

No output is defined.

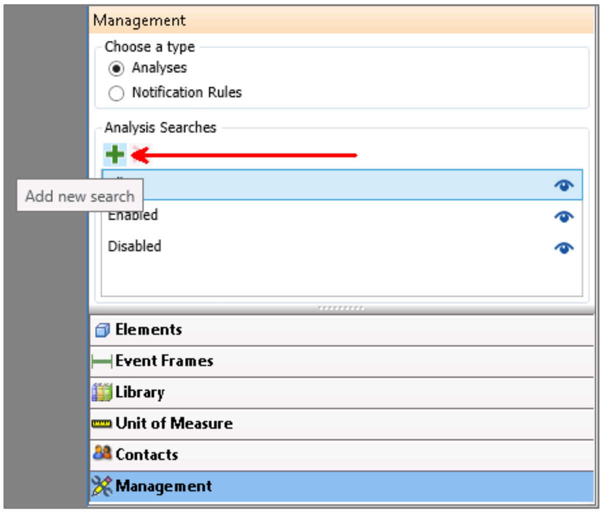
Figure 30. Service Time Calc analysis in error state

Why is this relevant for us?

If analyses are enabled before they are configured correctly, **the PI Analysis Service will automatically try to restart them every 15 minutes, causing unnecessary strain**.

In order to disable all the analyses in error state, proceed as follows:

1. Go to the Management tab.
2. Click on the ➕ symbol to add a new search (Figure 31)



Management

Choose a type
- ● Analyses
- ○ Notification Rules

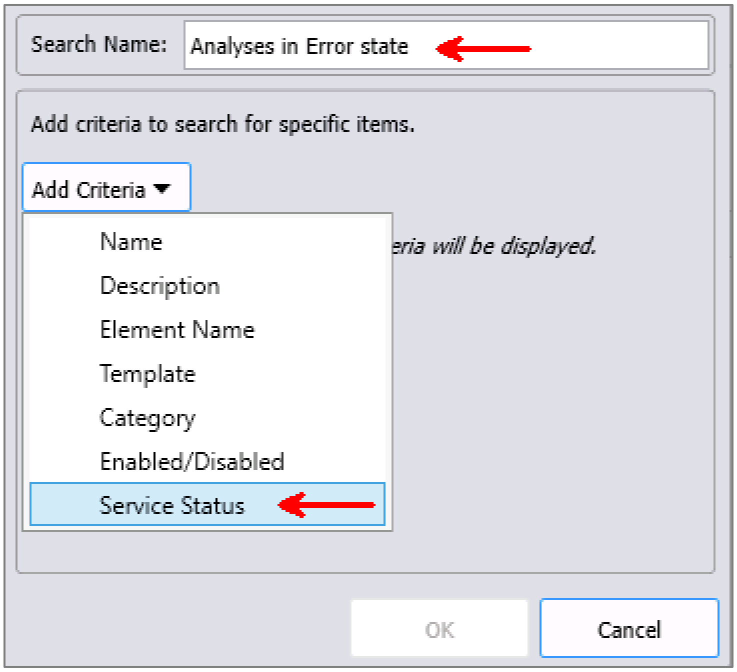
Analysis Searches

Add new search

- Enabled
- Disabled

Figure 31. Add new search under Management tab

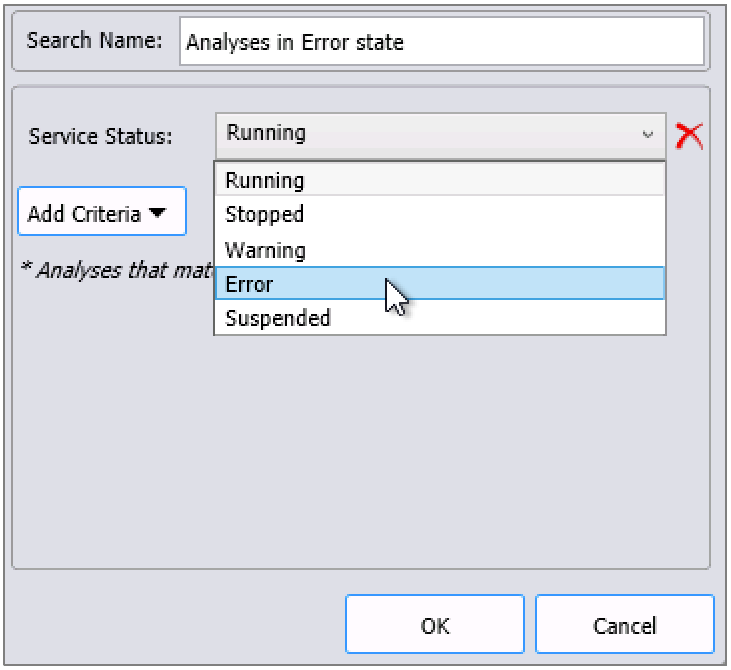
3. Name your search **Analyses in Error state** (Figure 32)
4. Add the filter criterion **Service Status** (Figure 32)


Search Name: Analyses in Error state

Add criteria to search for specific items.

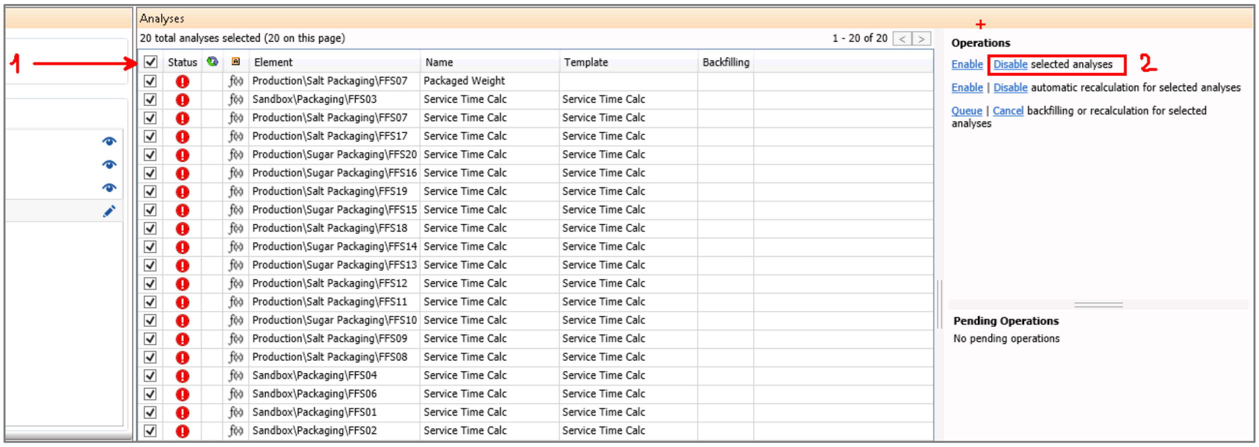
Figure 32. Search configuration

5. Select the **Error** status and then click OK (Figure 32)


Search Name: Analyses in Error state

Service Status: Running

6. Select all the filtered analyses using the first checkbox on the top (Figure 33)
7. Disable the selected analyses (Figure 33)



Analyses

20 total analyses selected (20 on this page)   1 - 20 of 20

☑	Status			Element	Name	Template	Backfilling
☑	❗		f(x)	Production\Salt Packaging\FFS07	Packaged Weight		
☑	❗		f(x)	Sandbox\Packaging\FFS03	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS07	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS17	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS20	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS16	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS19	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS15	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS18	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS14	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS13	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS12	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS11	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Sugar Packaging\FFS10	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS09	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Production\Salt Packaging\FFS08	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Sandbox\Packaging\FFS04	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Sandbox\Packaging\FFS06	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Sandbox\Packaging\FFS01	Service Time Calc	Service Time Calc	
☑	❗		f(x)	Sandbox\Packaging\FFS02	Service Time Calc	Service Time Calc	

Operations

Enable | Disable selected analyses

Enable | Disable automatic recalculation for selected analyses

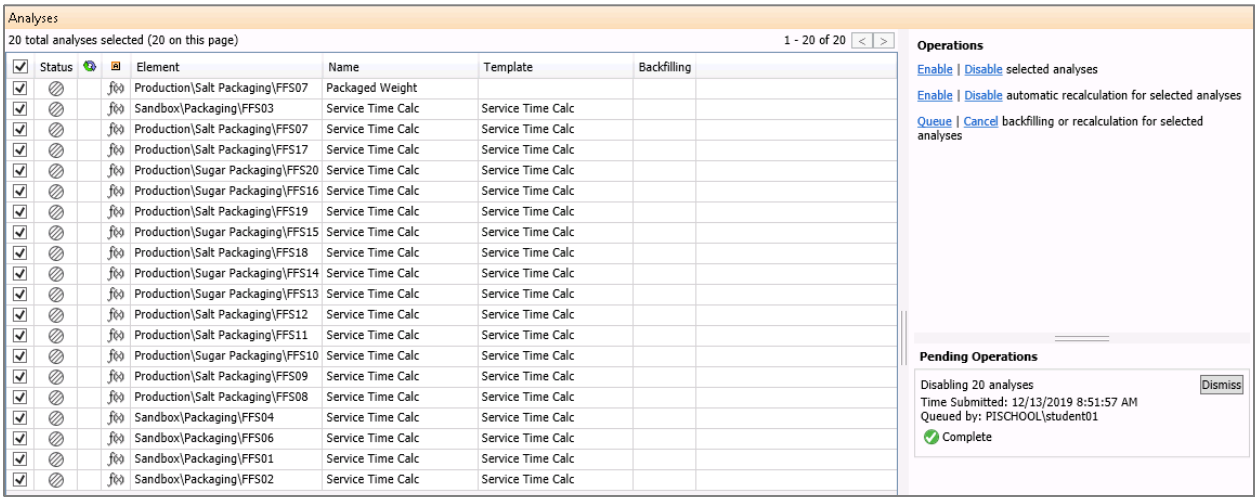
Queue | Cancel backfilling or recalculation for selected analyses

Pending Operations

No pending operations

Figure 33. Select and disable the analyses in error

The analyses should now display as disabled ⊘ (Figure 34).



Analyses

20 total analyses selected (20 on this page)   1 - 20 of 20

☑	Status			Element	Name	Template	Backfilling
☑	⊘		f(x)	Production\Salt Packaging\FFS07	Packaged Weight		
☑	⊘		f(x)	Sandbox\Packaging\FFS03	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS07	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS17	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS20	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS16	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS19	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS15	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS18	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS14	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS13	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS12	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS11	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Sugar Packaging\FFS10	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS09	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Production\Salt Packaging\FFS08	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Sandbox\Packaging\FFS04	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Sandbox\Packaging\FFS06	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Sandbox\Packaging\FFS01	Service Time Calc	Service Time Calc	
☑	⊘		f(x)	Sandbox\Packaging\FFS02	Service Time Calc	Service Time Calc	

Operations

Enable | Disable selected analyses

Enable | Disable automatic recalculation for selected analyses

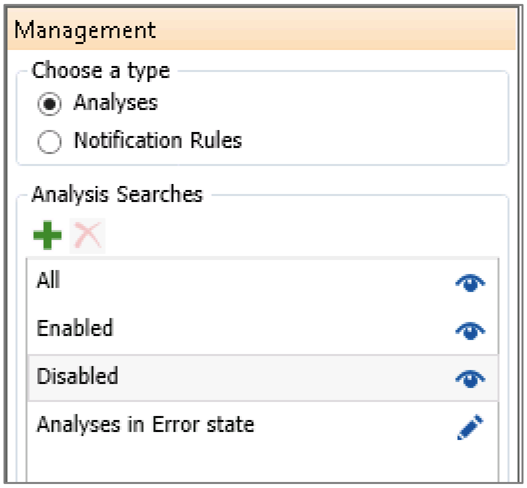
Queue | Cancel backfilling or recalculation for selected analyses

Pending Operations

Disabling 20 analyses   Dismiss
Time Submitted: 12/13/2019 8:51:57 AM
Queued by: PISCHOOL\student01
✓ Complete

Figure 34. Analyses in Disabled state

Should you want to view all the disabled analyses (for example when cleaning up your database from analyses that are no longer needed), you can use the default search option “Disabled” (Figure 35).


Management

Choose a type

● Analyses

○ Notification Rules

Analysis Searches

All

Enabled

Disabled

Analyses in Error state

Figure 35. Disabled analyses filter

A detailed example of the impact of analyses in warning or error states on the PI Analysis Service is provided in the PI Square post Asset Analytics Best Practices - Part 4: Analyses in Warning or Error (11).

✓ Best Practice	Disable all analyses in Warning or Error state until their configuration is fixed.

The work you have done

Remember all the issues we encountered when exploring the PI Vision displays in Section 2.3?

Let's recap what the issues were and how we solved them. Give yourself a big pat on your back for your hard work!

Issue # 1. Asset FFS05 is not available in the PI Vision Asset Dropdown Menu

Solution # 1. FFS05 was not using an AF element template. Once we assigned it a template, the asset became available in PI Vision.

Best Practice # 1. Always use **AF templates** to make PI Vision displays reusable!

Issue # 2. Trends and statistics for FFS units 3,4 and 6 are identical.

Solution # 2. FFS03, FFS04 and FFS06 have an additional attribute (Sugar Grain Size) configured using the "Allow Extensions" option, and a typo was made in the attribute configuration. We used a Derived Template for these FFS units, and this fixed the issue.

Best Practice # 2. Derived Templates, sometimes in combination with the **Excluded** attribute property, is the way to go to ensure consistency in your Asset Framework experience.

Issue # 3. The attribute Sealer temperature shows values outside the expected range.

Solution # 3. The Source Unit of Measure of the Sealer temperature had to be adjusted to reflect the UOM of the raw data stream.

Best Practice # 3. Asset Framework can automatically handle **UOM conversions** for you, make good use of this feature and don't forget to set the Source UOM explicitly.

Issue # 4. The attribute Last Service Date slows down the loading of the PI Vision display.

Solution # 4. The externally linked AF Table used by the Last Service Date attributes had a large number of rows. We parametrized the query and this greatly accelerated the data retrieval time.

Best Practice # 4. Parametrized queries are your friends when retrieving data from large tables in external databases.

Issue # 5. Asset FFS04 shows "No Data" for the Nominal Max Throughput.

Solution # 5. The Model attribute was using a static (<*none*>) data reference. We used an Enumeration Set and set each FFS unit's model choosing from a dropdown list.

Best Practice # 5. Enumeration Set help avoiding typos and ensuring consistent nomenclature throughout the database.

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset.

Solution # 6. The title used in the display was a static text box. We used a String Builder attribute which dynamically points to the name of the AF element.

Best Practice # 6. String Builder data reference and substitution parameters are very powerful tools for handling string values and creating dynamic pointers.

Finally, we saw how **AF Categories** help us organize, manage and filter AF objects such as attributes, analyses and elements.

Concerning **Asset Analytics**, we discussed:

1. the usefulness of **expression variables**. In our example, we used an expression variable to store the result of a TagAvg() expression. Expression variables improve the performance of the PI Analysis Service as well as simplify the analysis logic;

2. the importance of setting the **analysis scheduling** having in mind the required update frequency and the workload on the PI Analysis Service;

3. the advantage of **Rollup** calculations available in Asset Analytics compared to other solutions. Rollups automatically detect any modifications to the input attributes at execution time and update their behavior accordingly;

4. the choice between **consolidating or splitting** expressions into analyses depending on Dependency, Scheduling and Manageability;

5. the reason why analyses in **Warning or Error state** should be disabled until they are fixed.

Goodbye!

This Lab hopefully provided you with some good tips on how to improve your data infrastructure in Asset Framework and Asset Analytics.

For time constraints, we left out several aspects and features of Asset Analytics, such as Event Frame Generation, backfilling versus recalculation, and the Analysis data reference, to name just a few. We recommend exploring the **References** and the **Additional Resources** Sections below to keep learning and perfect the Noble Art of Asset Framework and Asset Analytics!

We would like to warmly thank you for your participation, and we hope to see you at the next PI World!

References

1. OSIsoft. A Journey from Historian to Infrastructure. [Online] 2015. [Cited: 12 12, 2019.] https://www.osisoft.com/White-Papers/A-Journey-from-Historian-to-Infrastructure/.

2. Top 10 Reasons to Love Your Modern PI Server. [Online] 2016. [Cited: 12 12, 2019.] https://www.osisoft.fr/white-papers/top-10-reasons-to-love-your-modern-pi-server/.

3. Global Sources. [Online] Global Sources. [Cited: 01 31, 2020.] https://www.globalsources.com/gsol/I/Powder-packing/p/sm/1163559821.htm#1163559821.

4. OSisoft. How AFTable Caching Works. [Online] 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB00539.

5. OSIsoft. Asset Analytics Best Practices - Part 3: Input Attributes. [Online] 4 22, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/04/22/asset-analytics-best-practices-part-3-input-attributes.

6. —. PI Analysis Service Best Practices. [Online] 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB01641.

7. The Analysis Service Data Cache. [Online] 2 28, 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB01495.

8. OSIsoft. Asset Analytics Best Practices - Part 1: Use Variables. [Online] 3 21, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/03/21/asset-analytics-best-practices-part-1-use-variables.

9. —. Asset Analytics Best Practices - Part 2: Data Density and Data Pattern. [Online] 03 28, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/03/28/asset-analytics-best-practices-part-2-data-density-and-data-pattern.

10. —. Asset Analytics Best Practices - Part 5: Scheduling. [Online] 7 11, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/07/11/asset-analytics-best-practices-part-5-scheduling.

11. —. Asset Analytics Best Practices - Part 4: Analyses in Warning or Error. [Online] 5 29, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/05/29/asset-analytics-best-practices-part-4-analyses-in-warning-or-error.

Additional resources

- **LiveLibrary**
- **OSIsoft Learning**
- Public class: "**Building PI System Assets and Analytics with AF**" (course material freely downloadable at the link "**Download Course Content**" at the bottom of the page).

Have an idea how to improve our products?

OSIsoft wants to hear from you!

https://feedback.osisoft.com/

OSIsoft. Learning

PI SYSTEM LEARNING MADE EASY!

Accelerate success with the new OSIsoft Learning platform.

VISIT LEARNING.OSISOFT.COM

© Copyright 2020

OSIsoft, LLC